



TECHNICKÁ UNIVERZITA V LIBERCI  
[www.tul.cz](http://www.tul.cz)



Fakulta mechatroniky, informatiky a mezioborových studií



# Diplomová práce

Liberec 2010/2011

**Bc. Tomáš Masopust**



# TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N 2612 – Elektrotechnika a informatika

Obor: 1802T007 – Informační technologie

## Vytvoření simulačního softwaru řešícího úlohy z teorie hromadné obsluhy s využitím metody Monte Carlo

The Creation of Simulation Software Solving Tasks of the  
Queueing Theory Using Monte Carlo Method

Diplomová práce

**Autor:** Bc. Tomáš Masopust

**Vedoucí DP:** Ing. Josef Chudoba, Ph.D.

**Rozsah práce:**

stran	slov	obrázků	grafů	tabulek	pramenů	příloh
76	12 586	7	11	9	30	1

V Liberci dne 18. 5. 2011



# TECHNICKÁ UNIVERZITA V LIBERCI

**Fakulta mechatroniky, informatiky a mezioborových studií**

Ústav nových technologií a aplikované informatiky

Akademický rok: 2010/11

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení: **Bc. Tomáš Masopust**

Studijní program: N 2612 – Elektrotechnika a informatika

Obor: 1802T007 – Informační technologie

Vedoucí ústavu Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto diplomovou práci:

Název tématu:

**Vytvoření simulačního softwaru řešící úlohy z teorie hromadné obsluhy s využitím metody Monte Carlo**

**The Creation of Simulation Software Solving Tasks of the Queueing Theory Using Monte Carlo Method**

Zásady pro vypracování:

1. Seznámení se s teorií hromadné obsluhy (SHO) s uvedením příkladů, které lze řešit.  
Recenze existujících softwarových nástrojů.
2. Popis metody Monte Carlo, uvedení výhod a nevýhod pro řešení úloh z teorie hromadné obsluhy.



3. Vytvoření softwarové aplikace např. v Matlabu, která simuluje úlohy z SHO.  
Software bude umožňovat modelování i neexponenciálních rozdělení doby do příchodu/odchodu nových jednotek.
4. Vytvoření aplikačních příkladů vhodných pro výuku.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 40-60 stran

Seznam odborné literatury:

- [1] Rálek P., Novák J., Chudoba J.: *Metody užívané v logistice*, Liberec: skriptu TUL, 2009.
- [2] Virius M.: *Aplikace matematické statistiky: metoda Monte Carlo*, Praha: skriptu ČVUT 1985, 1992, 1998. 168s. ISBN 80-01-01779-6.
- [3] Fabian F., Kluiber Z.: *Metoda Monte Carlo a možnosti jejího uplatnění*, Praha: PROSPEKTRUM s.r.o., 1998. 148s. ISBN 80-7175-058-1.
- [4] Peško Š.: *Operační analýza 2*, skriptu VŠŽ, Žilina 2001
- [5] Unčovský L.: *Stochastické modely operační analýzy*, Bratislava: Alfa Bratislava, 1. Vyd., 1980.
- [6] Zaplatílek K., Doňar B.: *Matlab – tvorba uživatelských aplikací*, Praha: BEN – technická literatura, 2005. 216 s. ISBN 80-7300-133-0.
- [7] Bartko R., Miller M.: *Matlab I. – algoritmizácia a riešenie úloh*, Trenčín: DIGITAL GRAPHICS, 2002. 288s. ISBN 80-968337-3-1.

Vedoucí diplomové práce: Ing. Josef Chudoba, Ph.D.



## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

V Liberci dne 18. 5. 2011

.....  
podpis



## **Poděkování**

Děkuji Ing. Josefu Chudobovi, Ph.D. za jeho odborné vedení, cenné připomínky a rady, které mi byly velkým přínosem při zpracování diplomové práce.

Dále bych chtěl poděkovat své rodině za morální a finanční podporu během celého mého studia.



## Abstrakt

Cílem této diplomové práce je vytvoření simulačního softwaru řešícího úlohy z teorie hromadné obsluhy s využitím metody Monte Carlo.

Výsledný software bude sloužit jako výuková pomůcka pro studenty předmětu MUL (Metody užívané v logistice) na Fakultě mechatroniky, informatiky a mezioborových studií na Technické univerzitě v Liberci.

Teoretická část práce se zabývá podrobným popisem systémů hromadné obsluhy a metody Monte Carlo, která využívá generátory náhodných čísel.

Praktická část se věnuje vytvoření simulačního softwaru a popisem jeho funkcí. Výsledný software s názvem SHO-MMC Simulator byl vytvořen v Matlabu. Disponuje grafickým uživatelským rozhraním pro zadávání vstupů a vyhodnocování výsledků. SHO-MMC Simulator poskytuje základní charakteristiky simulovaných systémů (střední počet požadavků v systému/frontě, střední doba, kterou požadavek stráví v systému/frontě atd.). Dále zobrazuje grafy okamžitých hodnot těchto charakteristik v průběhu doby simulace.

**Klíčová slova:** systémy hromadné obsluhy, teorie front, metoda Monte Carlo, generování náhodných čísel, simulace, Matlab



## Abstract

Goal of this graduate thesis is to create a simulation software solving the queueing theory with the Monte Carlo method.

The software will be used as a tool for MUL students (Methods used in logistic) at the Faculty of Mechatronics, Informatics and Interdisciplinary Studies at the Technical University in Liberec.

The theoretical part of this thesis consists of detailed description of different queueing systems and the Monte Carlo method which uses generators of random numbers.

The practical part describes the simulation software developing and it's functions. The final software is called SHO-MMC Simulator and was created in Matlab. Software provides a graphical user interface for data input entering and showing evaluation results. Basic characteristics of simulated systems (mean number of requests in the service/queue, mean waiting time in the service/queue, etc.) are also provided by SHO-MMC Simulator. Further it includes graphs of actual values of these characteristics during the simulation period.

**Keywords:** queueing systems, queueing theory, Monte Carlo method, random number generation, simulation, Matlab





## Obsah

<b>Úvod</b>	<b>11</b>
<b>1 Teorie systémů hromadné obsluhy</b>	<b>12</b>
1.1 Klasifikace SHO	14
1.1.1 Obecná	14
1.1.2 Vstupní tok	14
1.1.3 Fronta	15
1.1.4 Obslužný kanál	16
1.2 Kendallova klasifikace SHO	17
1.3 Základní charakteristiky SHO	19
1.3.1 Jednakanálový SHO $M/M/1/\infty/FIFO$	20
1.3.2 Vícekanálový SHO $M/M/x/\infty/FIFO$	22
1.4 Simulační softwarové nástroje	24
<b>2 Simulace metodou Monte Carlo</b>	<b>27</b>
2.1 Simulace	27
2.2 Metoda Monte Carlo	27
2.2.1 Historie	28
2.2.2 Přesnost metody Monte Carlo	29
2.3 Generátory náhodných čísel	29
2.3.1 Typy generátorů náhodných čísel	30
2.3.2 Generátory pseudonáhodných čísel	32
2.3.3 Požadavky na generátory a testování náhodnosti	34
2.3.4 Transformace náhodných čísel na hodnoty náhodných veličin	35
<b>3 Tvorba simulačního softwaru</b>	<b>37</b>
3.1 Požadavky k vytvoření softwaru	37
3.2 Programové prostředí Matlab	38
3.3 Transformace náhodných čísel na hodnoty náhodných veličin v prostředí Matlab	39
3.3.1 Transformace na exponenciální rozdělení	39



3.3.2	<i>Transformace na normální rozdělení</i>	40
3.4	Princip simulace	41
<b>4</b>	<b>Způsob použití softwaru SHO-MMC Simulator</b>	<b>43</b>
4.1	Způsob nainstalování a spuštění softwaru	43
4.2	Popis jednotlivých funkčních panelů softwaru	45
4.2.1	<i>Nastavení parametrů intervalů</i>	45
4.2.2	<i>Tabulka intervalů s parametry</i>	48
4.2.3	<i>Realizace</i>	48
4.2.4	<i>Vyhodnocení a zobrazení grafů</i>	49
4.3	Postup simulace	54
4.4	Ošetření výjimek	55
4.5	Nápověda	55
<b>5</b>	<b>Příklady použití softwaru</b>	<b>57</b>
5.1	Statické SHO	57
5.1.1	<i>Jednokanálový SHO</i>	57
5.1.2	<i>Vícekanálový SHO</i>	62
5.2	Dynamické SHO	65
5.2.1	<i>Změna počtu obslužných kanálů</i>	65
5.2.2	<i>Změna vstupního toku</i>	66
5.2.3	<i>Změna výstupního toku</i>	66
5.2.4	<i>Nenulové počáteční podmínky</i>	67
5.2.5	<i>Omezená velikost fronty</i>	68
<b>6</b>	<b>Závěr</b>	<b>69</b>
	<b>Literatura</b>	<b>70</b>
	<b>Seznam obrázků</b>	<b>73</b>
	<b>Seznam grafů</b>	<b>73</b>
	<b>Seznam tabulek</b>	<b>73</b>
	<b>Seznam symbolů a zkratk</b>	<b>74</b>
	<b>Seznam příloh</b>	<b>75</b>



## Úvod

Úkolem této diplomové práce je vytvoření simulačního softwaru řešícího úlohy z teorie hromadné obsluhy s využitím metody Monte Carlo.

Výsledný software bude sloužit jako výuková pomůcka pro studenty předmětu MUL (Metody užívané v logistice) na Fakultě mechatroniky, informatiky a mezioborových studií na Technické univerzitě v Liberci.

Software bude disponovat grafickým uživatelským rozhraním pro zadávání vstupů a vyhodnocování výsledků. Dokáže simulovat úlohy s exponenciálním, normálním a diskrétním rozdělením dob do příchodů a odchodů požadavků. Dále bude umožňovat simulaci jedno i více kanálových systémů hromadné obsluhy s omezením velikosti počtu požadavků čekajících ve frontě. Software bude simulovat jak ustálené, tak dynamicky se měnící systémy hromadné obsluhy.

Jako výsledky simulace bude software poskytovat základní charakteristiky systémů hromadné obsluhy (střední počet požadavků v systému/frontě, střední doba, kterou požadavek stráví v systému/frontě atd.). Dále bude zobrazovat v grafu okamžité hodnoty těchto charakteristik v průběhu doby simulace.

V jednotlivých kapitolách se čtenáři seznámí s teorií systémů hromadné obsluhy a s principem simulace pomocí metody Monte Carlo, která využívá generátory náhodných čísel.

Dále bude podrobně popsán způsob použití výsledného softwaru a vytvořeny aplikační příklady vhodné pro výuku.

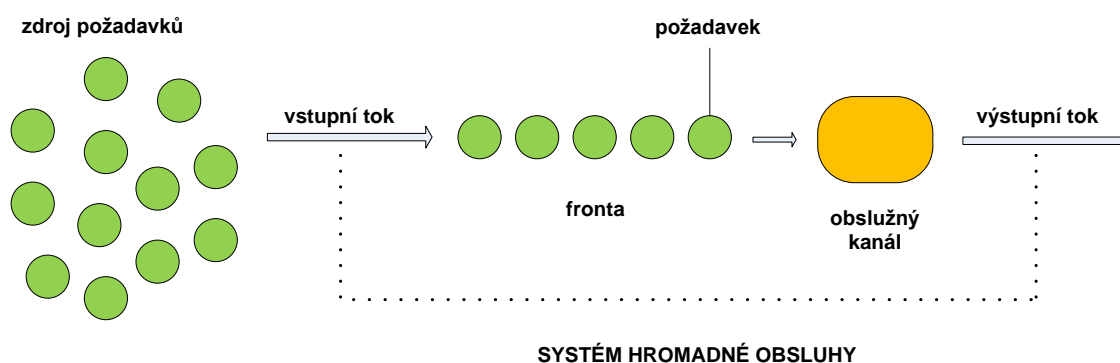
## 1 Teorie systémů hromadné obsluhy

Teorie systémů hromadné obsluhy (SHO) je technická (matematická) disciplína, která zkoumá činnost systémů, v nichž se opakovaně vyskytují toky požadavků, na kterých je třeba vykonat posloupnost operací (obsloužit je), tyto požadavky jsou co do vzniku a okamžiku výskytu zpravidla náhodné.

Základy teorii SHO položil začátkem 20. století dánský matematik Agner Krarup Erlang (1878-1929). Hlavním podnětem vzniku této disciplíny byl začínající rozvoj telefonních ústředěn, pro jejichž návrhy a dimenzování byla původně určena. Teorie SHO využívá teorii pravděpodobnosti, matematické statistiky a teorii náhodných funkcí [1].

Systém hromadné obsluhy (SHO) si můžeme představit jako systém, do kterého vstupují určité požadavky ze zdroje požadavků (zákazníci, výrobky, jednotky atd.), které jsou systémem uspokojeny (obslouženy). Z tohoto důvodu obsahuje systém hromadné obsluhy vstupní tok (toky), kterým požadavky do systému přicházejí a obslužný kanál (kanály), který se stará o obsloužení požadavku.

V některých případech může systém navíc obsahovat zařízení ke shromažďování požadavků čekajících na obsloužení, protože v okamžiku jejich vstupu do systému byl obslužný kanál zaneprázdněn obsluhou jiného požadavku. Toto zařízení se nazývá fronta, proto je také tato teorie někdy nazývána teorií front. Systém může obsahovat jednu či více front.



Obrázek 1: Obecné schéma systému hromadné obsluhy



Obecné schéma SHO představuje obrázek 1. Požadavky vstupují do systému s danou intenzitou vstupu (vstupní tok), kde jsou uspokojeny obslužným kanálem, který má také danou intenzitu obsluhy (výstupní tok). Pokud nejsou požadavky okamžitě po příchodu do systému obslouženy, řadí se do fronty a vyčkávají, dokud je obslužný kanál neuspokojí.

Typickým příkladem SHO jsou pokladny v supermarketu, kde vstupním tokem rozumíme náhodně přicházející zákazníky supermarketu a kanálem obsluhy pokladnu (pokladny), přičemž doba obslužení zákazníka je také náhodná. Fronta je tvořena zákazníky čekajícími na obsloužení. Výstupním tokem rozumíme obsloužené zákazníky opouštějící supermarket.

Cílem teorie SHO je optimalizovat počet obslužných kanálů tak, aby se netvořily příliš velké fronty a naopak aby byly všechny kanály obsluhy plně využity a nevznikaly zbytečné náklady na provoz systému.

Při modelování systémů hromadné obsluhy je možné použít dvě metody:

**Analytická** – spočívá v tom, že na základě známých funkčních vztahů a parametrů zkoumaného systému (modelu) pomocí nástrojů teorie pravděpodobnosti spočteme či odhadneme ty parametry systému, které nás zajímají (např. střední počet požadavků čekající ve frontě). Použitím analytických metod získáme přesné výsledky při zjišťování charakteristik systému, ale jejich zjišťování může být výpočetně složité (často při použití matematicky složitých vstupních nebo výstupních toků prakticky neřešitelné). Nevýhodou těchto metod je, že se dají použít pouze na systémy, které splňují určité předpoklady (stabilní, konstantní vstupy a výstupy apod.).

**Simulační** – využívá známých parametrů zkoumaného systému (modelu) k nasimulování dané situace pomocí specializovaného softwaru (SIMUL8, GOLDSIM, WITNESS a další), například použitím metody Monte Carlo. Tato simulace nám poskytne požadované parametry zkoumaného systému. Simulační metody se používají tam, kde nejsme schopni analyticky zjistit charakteristiky zkoumaných systémů.



## 1.1 Klasifikace SHO

### 1.1.1 Obecná

Na nejvyšší úrovni lze SHO dělit na *otevřené* a *uzavřené*. U otevřeného SHO se požadavky po jejich obslužení již do systému nevracejí. Naopak je tomu u uzavřeného SHO, kde se obslužené požadavky opět vracejí do zdroje požadavků systému (údržba provozních zařízení, mytí nádobí v jídelně) [1].

S tím úzce souvisí rozdělení SHO podle zdroje požadavků na *konečný* (kde je množina potencionálních požadavků ohraničena) a *nekonečný* (množina ohraničena není). Není efektivní uvažovat SHO s nekonečnou množinou požadavků za uzavřený (dostatečně velkou množinu lze považovat za nekonečně dlouhou – počet zákazníků v supermarketu) [1].

SHO lze dále dělit na systém *se ztrátami* a *beze ztrát*. Ztrátou rozumíme odmítnutí příchozího požadavku nebo jeho odchod ze systému bez toho, aniž by byl obslužen (zákazník v restauraci, plné parkoviště). U systému beze ztrát taková situace nastat nemůže, každý příchozí požadavek je systémem obslužen (počet pacientů na pohotovosti v nemocnici) [1], [2].

### 1.1.2 Vstupní tok

Vstupním tokem označujeme proces, při němž vznikají požadavky na kanál obsluhy. Vstupní toky mohou být [1], [3]:

**deterministické** – požadavky přicházejí k obslužnému kanálu v pravidelných a známých intervalech (periodický posun výrobní linky, periodický interval jízdy mezi autobusy).

**stochastické** – příchod požadavků k obslužnému kanálu není pravidelný, okamžiky příchodů jsou zcela náhodné. Časové intervaly mezi příchody jsou v těchto případech popisovány pomocí některého z pravděpodobnostních rozdělení (exponenciální, normální a další). Příkladem může být doba obsluhy zákazníků v obchodě.

**kombinovaný** – je kombinací obou výše zmiňovaných vstupních toků. To znamená, že některé požadavky přichází v pravidelných časových intervalech a některé



v proměnlivých. Pod tímto typem si můžeme představit výrobní linku, kde časový interval kolísá v určitých mezích, např.  $\pm 10$  sekund. Obdobně pro periodu příjezdu autobusů, kdy je rozmezí např.  $\pm 2$  minuty.

Vstupní tok dále rozlišujeme podle toho, jestli je realizován jednotlivými požadavky, či po skupinách. V případě skupinových vstupů lze dále specifikovat velikost těchto skupin, její závislost na typu požadavku atd. [1].

Podle typu požadavků rozdělujeme dva typy vstupů [3]:

**jednoduchý vstup** – všechny požadavky jsou obsluhovány stejným způsobem,

**složený vstup** – některé požadavky jsou obsluhovány podle jiného čekacího režimu.

### 1.1.3 Fronta

Způsob přechodu čekajících požadavků z fronty k obslužnému kanálu se nazývá režim fronty. Základními typy jsou [1][3]:

**fronta typu FIFO** (first-in/first-out) – požadavky jsou ve frontě řazeny podle jejich příchodů a v tomtéž pořadí jsou pak z fronty vybírány a obslouženy (nejdříve příchozí požadavek je nejdříve obsloužen). Typickým příkladem jsou všechny fronty zákazníků čekající na obslužení, ve kterých není dovoleno „přebíhání“ (supermarket, prodej vstupenek, pošta atd.).

**fronta typu LIFO** (last-in/first-out) – požadavky jsou ve frontě řazeny podle jejich příchodů, ale z fronty jsou vybírány v opačném pořadí (poslední příchozí požadavek je nejdříve obsloužen). Jako příklad lze uvést nakládku a vykládku kamiónů nebo skládání kontejnerů při přepravě jeden na druhý.

**fronta s prioritami PRI** – požadavky jsou ve frontě uspořádány podle své priority. Požadavky s nejvyšší prioritou budou nejdříve obslouženy. Příkladem mohou být naléhavé případy u lékaře, VIP vstupenky na určitou akci atd.

**náhodné pořadí SIRO** (selection in random order) – požadavky jsou do fronty řazeny nebo z ní vybírány zcela náhodně, případně podle daného rozložení pravděpodobnosti. S tímto typem fronty se lze setkat např. u výrobní linky, kde je vždy stejný pracovní



úkon prováděn na součástkách stejného typu, které se hromadí v zásobníku, ze kterého jsou náhodně vybírány.

**požadavky se řadí obecně GR** (general random)

Frontu můžeme dále dělit podle času, který je v ní ochoten požadavek strávit, tj. podle disciplíny fronty [3]:

**absolutně netrpělivá** – pokud jsou všechny kanály obsluhy obsazené, tak požadavek rezignuje na obsluhu a opouští systém neobsloužen (např. plné parkoviště, plná restaurace atd.).

**bez netrpělivosti** – požadavek čeká ve frontě neomezeně dlouhou dobu až do okamžiku, kdy je obsloužen (např. podání daňového přiznání, čekání na benzínce s prázdnou nádrží atd.).

**částečně netrpělivá** – požadavek je ochotný čekat ve frontě pouze určitou dobu, po jejím uplynutí opouští systém neobsloužen (typicky obchody, restaurace, kde je na zákazníkově si určit, jakou dobu je ochoten čekat).

#### 1.1.4 Obslužný kanál

Podle doby trvání obsluhy můžeme obslužné kanály rozdělit na kanály [1]:

**s pevnou dobou obsluhy** – doba trvání obsluhy jednoho požadavku je konstantní (výrobní linka).

**s náhodnou dobou obsluhy** – doba obsluhy jednoho požadavku je náhodná (kolísá) a je popsána některým rozdělením pravděpodobnosti (exponenciální, normální a další). Jako příklad lze uvést pokladní v supermarketu.

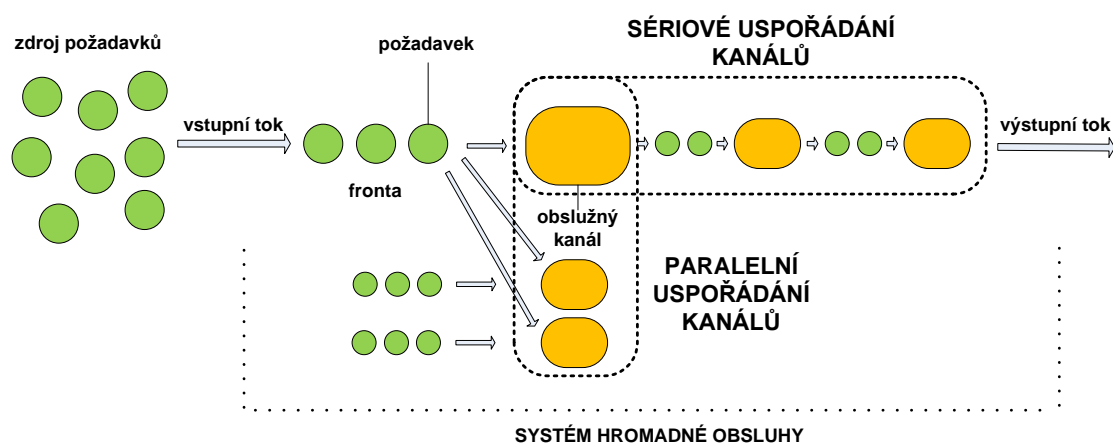
Kanál obsluhy nemusí být jen jeden, ale může jich být více. V případě více kanálů obsluhy je podstatné jejich uspořádání, to může být *sériové* nebo *paralelní*. Obecné Schéma SHO s uspořádáním obslužných kanálů je zobrazeno na obrázku 2.

Při paralelním uspořádání obslužných kanálů může být požadavek obsloužen kterýmkoli obslužným kanálem. U paralelního uspořádání může mít systém jednu



celkovou frontu pro všechny obslužné kanály, nebo samostatnou frontu pro každý obslužný kanál [3].

Pokud jsou obslužné kanály uspořádány sériově, musí požadavek postupně všemi těmito obslužnými kanály projít, teprve potom opouští systém. U sériového uspořádání obslužných kanálů se může v systému fronta tvořit pouze před prvním obslužným kanálem nebo systém může mít samostatnou frontu před každým obslužným kanálem [3].



Obrázek 2: Obecné schéma SHO s uspořádáním obslužných kanálů

## 1.2 Kendallová klasifikace SHO

Vzhledem k rozmanitosti systémů hromadné obsluhy je velice obtížné zavést symbolický popis, který by obsáhl všechny možnosti z výše uvedené klasifikace SHO. V roce 1963 David G. Kendall, britský statistik působící na univerzitách v Oxfordu a Cambridgi, představil svoji klasifikaci SHO. Tato Kendallová klasifikace popisuje SHO pomocí pěti, níže uvedených, základních charakteristik:

$$A/B/x/y/Z$$

kde:

**A** – popisuje charakter pravděpodobnostního rozdělení vstupního toku, tj. typ rozdělení s parametry charakterizující intervaly mezi příchody požadavků do systému



**B** – označuje charakter pravděpodobnostního rozdělení obslužného kanálu, tj. typ rozdělení s parametry charakterizující dobu trvání obsluhy požadavku

Kendall zavedl šest níže uvedených symbolů pro běžně používané typy rozdělení intervalů vstupního toku a obslužného kanálu (znaky A, B).

- **M** – Markovský-Poissonovský vstupní/výstupní tok, exponenciální rozdělení vzájemně nezávislých intervalů příchodů požadavků či doby obsluhy (doba do poruchy běžného zařízení)
- **E<sub>k</sub>** – Erlangovo rozdělení intervalů s  $k \in N$  stupni volnosti (doba do k-poruchy zařízení, je-li nahrazeno po poruše novým)
- **K<sub>n</sub>** –  $\chi^2$  rozdělení s  $n$  stupni volnosti
- **D** – konstantní intervaly (výrobní linka)
- **N** – normální rozdělení intervalů (výrobní linka)
- **G** – obecné, tj. jakékoliv rozložení intervalů příchodů a dob obsluhy

**x** – číslo udávající počet paralelně uspořádaných obslužných kanálů

**y** – představuje celkovou kapacitu systému hromadné obsluhy, tj. počet požadavků, které mohou být v systému najednou přítomny (požadavky v obsluze + požadavky čekající ve frontě). Obvykle se předpokládá, že celková kapacita systému může být nekonečná ( $\infty$ ).

**Z** – udává režim fronty (FIFO, LIFO, PRI, SIRO, GR)

Kendallova klasifikace nepostihuje všechny možné typy SHO. Původní Kendallova klasifikace obsahovala pouze tři základní údaje A/B/x, tj. charakter vstupního toku a obslužného kanálu a počet paralelně řazených kanálů s tím, že byl předpoklad otevřeného systému beze ztrát s frontou FIFO.

Pro popis speciálních systémů je třeba Kendallovu klasifikaci rozšířit. Pokud je potřeba popis systému rozšířit například o údaj o početnosti zdroje požadavků, přidá se před symbol Z další atribut. Není-li uveden, předpokládá se neomezený zdroj požadavků. Další rozšíření popisu systému je možné přidáním doplňujících symbolů s vysvětlením, co představují [1], [2].



Kendallova klasifikace s příklady je velice podrobně popsána a vysvětlena v literatuře [16], [17]. V rámci této diplomové práce bylo kromě toho využito i literatury [1], [2], [3].

Příklad systému hromadné obsluhy zapsaného pomocí Kendallové klasifikace může vypadat následovně:

M/M/4/50/FIFO

Jedná se o systém hromadné obsluhy s exponenciálním rozdělením intervalů mezi příchody požadavků i doby obsluhy jednotlivých požadavků. Systém disponuje čtyřmi obslužnými kanály. Celková kapacita systému je 50 požadavků. Režim fronty je FIFO, maximální velikost fronty je 46 požadavků (50-4). Parametr udávající zdroj požadavků není uveden, předpokládá se tedy, že zdroj požadavků není omezen.

### 1.3 Základní charakteristiky SHO

Základními charakteristikami systémů hromadné obsluhy, které budeme chtít zjistit, jsou nejčastěji [3]:

- |   |                  |
|---|------------------|
| ○ střední počet požadavků v systému               | $\overline{n_s}$ |
| ○ střední počet požadavků ve frontě               | $\overline{n_f}$ |
| ○ střední doba, kterou požadavek stráví v systému | $\overline{t_s}$ |
| ○ střední doba, kterou požadavek stráví ve frontě | $\overline{t_f}$ |

Dalšími charakteristikami, které můžeme u systému hromadné obsluhy zjišťovat, jsou například pravděpodobnosti, že se v systému nebo ve frontě nachází právě  $n$  požadavků. Dále pravděpodobnost, že bude požadavek po vstupu do systému okamžitě obsloužen, tj. požadavek bude obsloužen bez čekání nebo pravděpodobnost, že bude požadavek čekat v systému jen určitou dobu atd.

Abychom mohli zjistit výše uvedené charakteristiky SHO, je potřeba zjistit pravděpodobnosti jednotlivých stavů systému  $\mathbf{p} = (p_0, p_1, p_2 \dots p_n)$ . Kde  $p_0$  představuje pravděpodobnost, že se systém nachází ve stavu 0, tj. v systému není žádný požadavek,



$p_1$  je pravděpodobnost, že se v systému nachází právě jeden požadavek,  $p_2$  je pravděpodobnost, že jsou v systému právě dva požadavky atd.

Pravděpodobnosti jednotlivých stavů systému hromadné obsluhy lze zjistit analyticky nebo pomocí simulace.

### 1.3.1 Jednokanálový SHO M/M/1/ $\infty$ /FIFO

Jedná se o jednokanálový SHO s exponenciálním rozdělením intervalů mezi příchody požadavků i doby obsluhy jednotlivých požadavků. Kapacita systému není omezená, systém disponuje frontou s režimem FIFO.

Pro všechny vstupy a výstupy platí předpoklad, že jsou konstantní a nezávislé na předchozích vstupech (resp. výstupech) [3].

Parametry systému:

$\lambda$  – střední intenzita vstupního toku (střední počet požadavků, které vstoupí do systému během dané časové jednotky  $s^{-1}$ ,  $min^{-1}$ ,  $hod^{-1}$ )

$\mu$  – střední intenzita výstupního toku (střední počet požadavků, které jsou systémem obslouženy během dané časové jednotky  $s^{-1}$ ,  $min^{-1}$ ,  $hod^{-1}$ )

$\rho$  – střední intenzita provozu  $\rho = \frac{\lambda}{\mu}$

Aby byl systém stabilní, musí splňovat podmínku  $\mu > \lambda$  neboli  $\rho < 1$  tj., že střední intenzita výstupního toku musí být větší, než střední intenzita vstupního toku. Pokud by tomu tak nebylo, systém by nestačil obsluhovat vstupující požadavky a fronta by narůstala do nekonečna nebo do maximální délky fronty.

Po splnění těchto podmínek můžeme vyjádřit obecné vztahy pro výpočet pravděpodobností jednotlivých stavů u ustálených systémů, které jsou následující:

$$p_n = \rho^n \cdot (1 - \rho), \quad n = 0, 1, 2 \dots \quad (1)$$

Ze známých pravděpodobností jednotlivých stavů lze již vyjádřit jednotlivé charakteristiky systému hromadné obsluhy.

**střední počet požadavků v systému**

$$\overline{n_s} = \sum_{n=1}^{\infty} n \cdot p_n \quad (2)$$

vztah pro ustálený systém

$$\overline{n_s} = \frac{\lambda}{\mu - \lambda} \quad (3)$$

**střední počet požadavků ve frontě**

Fronta v systému vzniká jen tehdy, pokud je obsazen obslužný kanál, tj. je-li v systému víc než jeden požadavek.

$$\overline{n_f} = \sum_{n=2}^{\infty} (n - 1) \cdot p_n \quad (4)$$

vztah pro ustálený systém

$$\overline{n_f} = \frac{\lambda^2}{\mu \cdot (\mu - \lambda)} \quad (5)$$

**střední doba, kterou požadavek stráví v systému**

$$\overline{t_s} = \sum_{n=1}^{\infty} \frac{n \cdot p_n}{\lambda} \quad (6)$$

vztah pro ustálený systém

$$\overline{t_s} = \frac{1}{\mu - \lambda} \quad (7)$$

**střední doba, kterou požadavek stráví ve frontě**

$$\overline{t_f} = \sum_{n=2}^{\infty} \frac{(n - 1) \cdot p_n}{\lambda} \quad (8)$$



vztah pro ustálený systém

$$\bar{t}_f = \frac{\lambda}{\mu \cdot (\mu - \lambda)} \quad (9)$$

### 1.3.2 Vícekanálový SHO M/M/x/∞/FIFO

Jedná se o vícekanálový SHO s exponenciálním rozdělením intervalů mezi příchody požadavků i doby obsluhy jednotlivých požadavků. Kapacita systému není omezená, systém disponuje frontou s režimem FIFO.

Pro všechny vstupy a výstupy platí předpoklad, že jsou konstantní a nezávislé na předchozích vstupech (resp. výstupech) [3].

Parametry systému:

$\lambda$  – střední intenzita vstupního toku (střední počet požadavků, které vstoupí do systému během dané časové jednotky  $s^{-1}$ ,  $min^{-1}$ ,  $hod^{-1}$ ),

$\mu_x$  – střední intenzita výstupního toku (střední počet požadavků, které jsou systémem obslouženy během dané časové jednotky  $s^{-1}$ ,  $min^{-1}$ ,  $hod^{-1}$ )

$\rho$  – střední intenzita provozu  $\rho = \frac{\lambda}{\mu}$ , kde  $\mu$  je střední intenzita výstupního toku jednoho obslužného kanálu

$x$  – počet obslužných kanálů

Aby byl systém stabilní, musí splňovat podmínku  $\mu_x > \lambda$  neboli  $\rho < 1$ , tj. že střední intenzita výstupního toku musí být větší než střední intenzita vstupního toku. Střední intenzitou výstupu  $\mu_x$  je myšlen součet středních intenzit výstupu jednotlivých kanálů, předpokládáme, že všechny kanály mají stejnou intenzitu výstupu  $\mu$  ( $\mu_x = x \cdot \mu$ ). Pokud by tomu tak nebylo, systém by nestačil obsluhovat vstupující požadavky a fronta by narůstala do nekonečna nebo do maximální délky fronty.



Jestliže systém splňuje tyto podmínky, můžeme vyjádřit obecné vztahy pro výpočet pravděpodobností jednotlivých stavů následovně:

$$p_0 = \frac{1}{\frac{\rho^x}{x! \left(1 - \frac{\rho}{x}\right)} + \sum_{n=0}^{x-1} \frac{\rho^n}{n!}} \quad (10)$$

$$p_n = \frac{\rho^n}{n!} \cdot p_0 \quad \text{pro } n = 1, 2, \dots, x \quad (11)$$

$$p_n = \frac{\rho^n}{x! \cdot x^{n-x}} \cdot p_0 \quad \text{pro } n > x \quad (12)$$

Ze známých pravděpodobností jednotlivých stavů lze již vyjádřit jednotlivé charakteristiky vícekanálového systému hromadné obsluhy.

#### **střední počet požadavků v systému**

$$\bar{n}_s = \sum_{n=1}^{\infty} n \cdot p_n \quad (13)$$

vztah pro ustálený systém

$$\bar{n}_s = \frac{\rho^{x+1}}{x \cdot x! \left(1 - \frac{\rho}{x}\right)^2} \cdot p_0 + \rho \quad (14)$$

#### **střední počet požadavků ve frontě**

Fronta v systému vzniká pouze tehdy, jsou-li obsazeny všechny obslužné kanály, tj. je-li v systému víc požadavků než obslužných kanálů.

$$\bar{n}_f = \sum_{n=2}^{\infty} (n-1) \cdot p_{n+x-1} \quad (15)$$

vztah pro ustálený systém

$$\bar{n}_f = \bar{n}_s - \rho = \frac{\rho^{x+1}}{x \cdot x! \left(1 - \frac{\rho}{x}\right)^2} \cdot p_0 \quad (16)$$



**střední doba, kterou požadavek stráví v systému**

$$\bar{t}_s = \sum_{n=1}^{\infty} \frac{n \cdot p_n}{\lambda} \quad (17)$$

vztah pro ustálený systém

$$\bar{t}_s = \frac{\bar{n}_s}{\lambda} \quad (18)$$

**střední doba, kterou požadavek stráví ve frontě**

$$\bar{t}_f = \sum_{n=2}^{\infty} \frac{(n-1) \cdot p_{n-1}}{\lambda} \quad (19)$$

vztah pro ustálený systém

$$\bar{t}_f = \frac{\bar{n}_f}{\lambda} \quad (20)$$

### **1.4 Simulační softwarové nástroje**

Specializovaných softwarových nástrojů, kterými je možné řešit a simulovat chování systémů hromadné obsluhy, je mnoho. Jedná se obecně o simulační softwary, které nemusí být určeny výhradně pro simulování SHO.

**SIMUL8** – softwarový balík SIMUL8 nabízí uživatelsky přívětivé, integrované prostředí, umožňující snadnou a rychlou práci s modely pro dynamickou diskrétní simulaci procesů. Pomocí programu SIMUL8 snadno vytvoříme přesné, flexibilní a robustní simulace podnikových, výrobních nebo logistických procesů či systémů ve velmi krátkém čase. Program SIMUL8 je produktem americké firmy Simul8 Corporation [25].

Ve verzi „Standard“ stojí 29 000,- Kč, ve verzi „Professional“ 97 000,- Kč

[www.simul8.com](http://www.simul8.com), [www.simul8.cz](http://www.simul8.cz)

**GOLDSIM** – je univerzální simulační software, využívající metody Monte Carlo pro dynamické modelování složitých systémů z oblasti obchodu, strojírenství a vědy.





Goldsim kvantitativně určuje nejistotu a míru rizika navrhovaných systémů, tím přináší podklady pro manažerská rozhodnutí [26].

Cena softwaru je \$3950

[www.goldsim.com](http://www.goldsim.com)

**ARENA** – software od společnosti Rockwell Automation je obecným simulačním jazykem především pro průmyslové aplikace a business process reengineering. Je součástí větší skupiny produktů ARENA pro podporu manažerského rozhodování [27].

Základní produkt „Arena Basic“ stojí \$2495

[www.arenasimulation.com](http://www.arenasimulation.com)

**WITNESS** – simulační software od společnosti Lanner Group. Je určen především pro simulaci a optimalizaci výrobních, obslužných a logistických systémů. Program je dodáván ve dvou variantách: *Manufacturing Performance Edition* pro simulaci výrobních procesů a *Service and Process Performance Edition* pro oblast služeb. Odlišnost těchto variant spočívá pouze v jiné terminologii a v jiné sadě předpřipravených prvků. Po funkční stránce jsou identické. Mezi rozšiřující doplňky patří například Witness Visio, který propojí Witness s Microsoft Visio produktem [28], [29].

Cena základního produktu stojí 720 000 Kč. Na českém trhu je distribuován firmou HUMUSOFT.

[www.lanner.com](http://www.lanner.com), [www.humusoft.com](http://www.humusoft.com)

**PROMODEL** – je simulační software určen k hodnocení, plánování a projektování výrobních, skladovacích, logistických a jiných operativních a strategických potřeb. Základním produktem je Process Simulator. Společnost má také v nabídce produkty, použitelné pro konkrétní oblasti ve společnosti, ServiceModel (služby), MedModel (zdravotní péče) a další. Program PROMODEL je produktem americké společnosti PROMODEL Corporation [30].

Cena produktu Process Simulator je \$4200

[www.promodel.com](http://www.promodel.com)



Z uvedených příkladů můžeme konstatovat, že náklady na pořízení specializovaných softwarových simulačních nástrojů jsou velmi vysoké. Protože se jedná o velmi sofistikovaný software, musí se investovat i do školení pracovníků, kteří budou s tímto softwarem pracovat.



## 2 Simulace metodou Monte Carlo

### 2.1 Simulace

Slovem simulace, zjednodušeně řečeno, rozumíme napodobování nějakého systému (věci, stavu, procesu) a jeho chování v reálném světě. V dalším textu budeme pojmem simulace rozumět počítačovou simulaci, tj. simulaci prováděnou za pomoci výpočetní techniky. Potom chápeme slovo simulace jako experiment s počítačovým modelem.

Vznik simulace jako nástroje ke zkoumání chování systémů sahá do období 2. světové války a její první využití bylo pro vojenské účely. První simulace byla použita v projektu Manhattan (projekt Manhattan byl krycí název pro utajený vývoj atomové bomby). Za pomoci metody Monte Carlo se simulovalo chování jaderných výbuchů [5].

V literatuře je možné nalézt několik definic simulace:

**Robert E. Shannon:** - Simulace je proces navrhování modelu reálného systému a provádění experimentů s tímto modelem za účelem pochopení systému nebo hodnocení různých strategií (se stanovenými kritérii) pro provoz systému.

**Thomas H. Naylor** - Simulace je numerická metoda, která spočívá v experimentování s matematickými modely reálných systémů na číslicových počítačích.

**Ole-Johan Dahl** – Simulace je technika, která nahrazuje dynamický systém modelem, na kterém se provádí experimenty s cílem získat informace o původním zkoumaném systému.

### 2.2 Metoda Monte Carlo

Metoda Monte Carlo je stochastická metoda pro řešení matematických a jiných problémů, využívající modelování náhodných veličin.

Základní myšlenka této metody je velice jednoduchá, chceme určit střední hodnotu veličiny, která je výsledkem náhodného děje. Simulace za pomoci metody



Monte Carlo probíhá tak, že pro reálný systém (nemusí se jednat o skutečně existující systém, nýbrž třeba teprve navrhovaný) vytvoříme jeho simulační model se stejnými pravděpodobnostními charakteristikami a chování reálného systému mnohonásobně simulujeme na vytvořeném simulačním modelu [6]. Využití metody Monte Carlo je opravdu velmi široké, používá se k řešení složitých problémů z oblasti techniky, ekonomiky, řízení dopravy, činnosti telefonních centrál a při řešení problémů v samotné matematice.

### 2.2.1 Historie

Historie vzniku metody Monte Carlo sahá až do roku 1777, její základní principy formuloval francouzský matematik, přírodovědec a astronom Georges Louis Leclerc de Buffon ve svém známém problému s jehlou, známý pod pojmem Buffonova úloha nebo Buffonova jehla.

Metoda jako taková pod názvem Monte Carlo vznikla v období 2. světové války. Jejími zakladateli byli Stanislaw Marcin Ulam a John von Neumann, kteří pracovali ve Spojených státech na vývoji atomové bomby.

Při výzkumu chování neutronů a jejich pohlcování různými materiály se objevil problém, jak určit počet neutronů, které proniknou například nádrží vody o určitých rozměrech. Nedařilo se nalézt praktickou metodu, která by předpovídala chování jednotlivých neutronů při průchodu sloupcem vody a náhodně se srážejících s atomy vodíku a kyslíku. Autoři využili k modelování „života neutronu“ techniku kola rulety (odtud název Monte Carlo). Vycházeli z toho, že k pohlcení neutronu jiným atomem dojde přibližně v jednom případě ze sta. Každé roztočení rulety (se sto dílky, z nichž jen jeden by představoval pohlcení neutronu) by simulovalo pohyb neutronu. V případě, že by se zastavila na dílku, který znázorňuje pohlcení neutronu, neutron by vodou neprošel. Toto by se opakovalo, dokud by neutron nebyl pohlcen nebo dokud by úspěšně neprošel vodním sloupcem.

Provádět tyto simulace ručně s použitím opravdové rulety by bylo časově nereálné, proto se k simulaci použila v té době již vyvíjená a dostupná, z dnešního pohledu jednoduchá, výpočetní technika. Po provedení velkého počtu těchto simulací



se podařilo získat celkem přesnou informaci o procentu neutronů, které prošly sloupcem vody, aniž by byly pohlceny [5].

### 2.2.2 Přesnost metody Monte Carlo

K odhadu chyby výsledku získaného metodou Monte Carlo se nejčastěji používá střední kvadratická chyba aritmetického průměru. Chyba výsledku získaného  $n$  pokusy obvykle klesá jako  $1/\sqrt{n}$  [6]. Proto k dosažení o řád vyšší přesnosti výsledků musíme provést alespoň o dva řády více pokusů (realizací). Ve vzorci  $n$  představuje počet provedených pokusů (realizací) metodou Monte Carlo.

### 2.3 Generátory náhodných čísel

Generátory náhodných čísel jsou zařízení nebo procedury, které generují čísla, která jsou opravdu náhodná nebo jen na první pohled jako náhodná čísla vypadají.

Pokud se zmíní pojem generátor náhodných čísel, většina lidí si pod tímto pojmem vybaví nejjednodušší představitele, se kterými se v životě běžně setkávají. Těmi mohou být losování čísel z osudí u Sportky, kolo na slevy v Mountfieldu nebo třeba hod hrací kostkou.

Generátory náhodných čísel se staly běžnou součástí našeho života, i když si to možná na první pohled neuvědomujeme. Můžeme je nalézt zejména v oblasti kryptografie, tedy zabezpečení elektronických dat, jako je elektronická pošta, internetové bankovníctví. Generátor náhodných čísel je zabudován také ve většině hudebních přehrávačů, kde plní funkci náhodného výběru skladeb. Využívá je i většina simulačních a modelačních softwarů. Někaký generátor je také implementován vesměs v každém programovacím jazyce.

Při pokusech vygenerovat hodnoty s náhodnými vlastnostmi se vyskytla řada teoretických i praktických problémů. Prvním bylo zodpovědět otázku, co to náhoda vůbec je a jestli je daný proces náhodný nebo není. Výsledkem jsou dva formulované závěry [9]:

- jestliže je proces generován pomocí nějakého algoritmu, potom nemůže být náhodný, jelikož jej lze za stejných podmínek zopakovat a dojít ke stejnému výsledku



- pokud je proces generován bez použití algoritmů (pomocí nějakého fyzikálního jevu), pak je otázka náhodnosti obecně nerozhodnutelná.

Z výše uvedených závěrů je možné říci, že není rozhodující, zda je proces náhodný nebo ne, ale jestli se jako náhodný chová. S tímto problémem úzce souvisí vývoj testů, podle kterých se rozhoduje, má-li daný proces vlastnosti odpovídající náhodnému procesu [9].

### 2.3.1 Typy generátorů náhodných čísel

Generátory náhodných čísel lze rozdělit na dva základní druhy. Prvním z nich je generátor PRNG (Pseudo-Random Number Generator), tedy generátor pseudonáhodných čísel. Druhým je generátor TRNG (True Random Number Generator), pro tento generátor můžeme v češtině použít termín Pravý generátor náhodných čísel nebo možná lépe vystihující označení Generátor opravdu náhodných čísel [10], [11].

Generátor opravdu náhodných čísel (TRNG) generuje náhodná čísla za pomoci fyzikálních jevů. S těmito generátory se můžeme setkat v hardwarové podobě. Typickým příkladem čistě mechanického generátoru je výše zmiňovaná ruleta nebo osudí Sportky. Ve většině případů se ale používá generátor kombinující hardwarovou část, která zpracovává nějaký fyzikální děj, a softwarovou část, která ze získaných hodnot provádí vlastní generování.

Generátor pseudonáhodných čísel (PRNG) je deterministický program, který generuje čísla na základě určitého algoritmu. Snahou je, aby se vygenerovaná čísla co nejvíce podobala skutečně náhodným číslům, tj. aby je nebylo možné statistickými testy rozeznat od náhodných [10], [11].

Podrobněji můžeme generátory náhodných čísel rozdělit na [9], [12]:

**tabulky náhodných čísel** – jsou vhodné pro výpočty malého rozsahu (Tippetové tabulky 40 000 čísel atd.). Po převedení do elektronické podoby mohou sloužit počítači jako vstupní data. Jejich výhodou jsou kvalitní data (co se náhodnosti týče), ale i rozsáhlejší tabulky jsou pro dnešní potřeby příliš malé. Ukázku takové tabulky můžeme vidět na obrázku 3.



73735	45963	78134	63873
02965	58303	90708	20025
98859	23851	27965	62394
33666	62570	64775	78428
81666	26440	20422	05720
15838	47174	76866	14330
89793	34378	08730	56522
78155	22466	81978	57323
16381	66207	11698	99314
75002	80827	53867	37797
99982	27601	62686	44711
84543	87442	50033	14021
77757	54043	46176	42391
80871	32792	87989	72248
30500	28220	12444	71840

Obrázek 3: Tabulka náhodných čísel [15]

**mechanické generátory** – jedná se o nejznámější generátory náhodných čísel. Typickým příkladem může být třeba již výše zmiňovaná ruleta, házení kostkou, mechanický systém na losování Sportky nebo také třeba hod mincí a další. Nevýhodou je, že tyto generátory jsou pro simulaci na počítači nepoužitelné.

**fyzikální generátory** – využívají fyzikálních dějů, které mají náhodný charakter. Například využití radioaktivního zdroje, časové okamžiky, ve kterých se radioaktivní zdroj rozpadá, jsou zcela nepředvídatelné. Dalšími jsou šumové generátory, kvantové generátory, lávové lampy apod. Do této skupiny se také dají zařadit generátory využívající jako prvek náhody vstup od uživatele (pohyb myši, prodleva mezi stiskem kláves atd.). Jejich největší výhodou je, že čísla z nich získaná se dají považovat za opravdu náhodná.

**aritmetické generátory** – nejrozšířenější typ generátorů používaných v počítačové simulaci. Náhodná čísla jsou tvořena aritmetickými operacemi pomocí rekurentních vztahů, to znamená, že následující číslo závisí na jednom nebo více předchozích číslech. Čísla získaná z těchto generátorů nazýváme pseudonáhodná, protože jsou vypočtená, nikoliv náhodná.

Dále se generátory náhodných čísel dělí na [9], [12]:

**primární generátory** – generují posloupnost náhodných čísel s rovnoměrným rozdělením.



**sekundární generátory** – transformují posloupnost náhodných čísel s rovnoměrným rozdělením na jiná požadovaná rozdělení.

### 2.3.2 Generátory pseudonáhodných čísel

Jak bylo výše uvedeno, generátor pseudonáhodných čísel je deterministický program, který generuje posloupnost čísel, které lze považovat za náhodná.

Nejstarším a nejznámějším generátorem pseudonáhodných čísel je lineární kongruenční generátor. Těmto generátorům se pro úvodní inicializaci zadávají vstupní data tzv. *random seed*, která se doplní do příslušného algoritmu a poté je provedeno vlastní generování. Random seed buď může zadávat uživatel, nebo je jeho výběr ponechán na programu (např. podle aktuálního času systémových hodin) [6], [9].

Protože se jedná o deterministické programy, které mají k dispozici jen omezené množství paměti, jsou posloupnosti takto vygenerovaných čísel nutně periodické, tedy po určité době se začnou opakovat. Proto také hovoříme o pseudonáhodných generátorech.

Nejznámější typy lineárních kongruenčních generátorů a jejich vztahy [12]:

#### **smíšený**

$$x_{n+1} = (a \cdot x_n + c) \pmod{M}$$

#### **multiplikativní**

$$x_{n+1} = (a \cdot x_n) \pmod{M}$$

#### **aditivní**

$$x_{n+1} = (x_n + x_{n-1}) \pmod{M}$$

$$\text{pro } 0 < M; 2 \leq a < M; 0 \leq c < M; 0 \leq x_0 < M$$

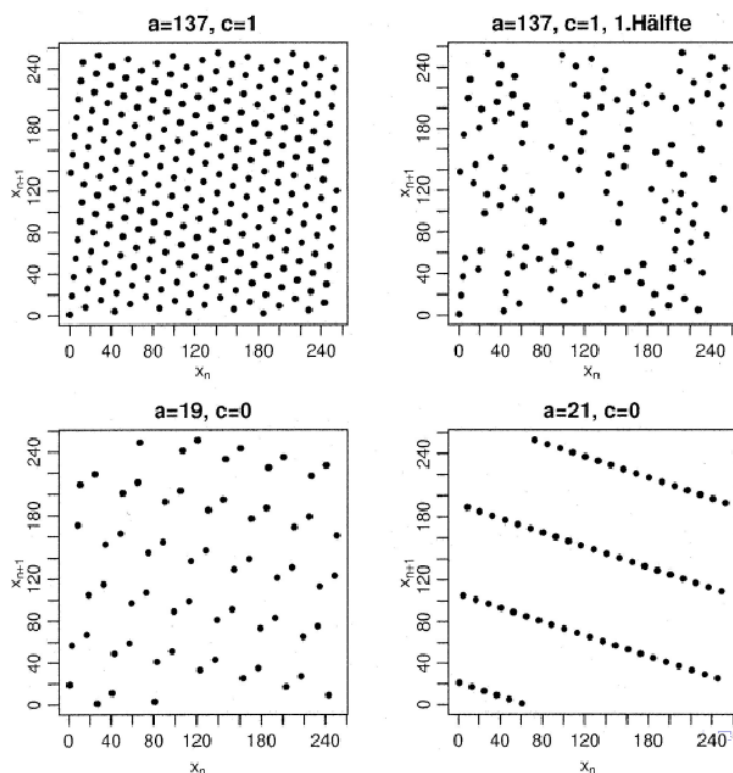
V těchto vztazích jsou **a** (násobitel), **c** (inkrement), **M** (modul) vhodně zvolené konstanty (různé podle typu generátoru). Výraz  $x_{n+1}$  je aktuálně generované číslo a  $x_n$  je číslo vygenerované v předchozím kroku (popř. ve více předchozích krocích  $x_{n-1}$ ). Hodnota prvního kroku  $x_0$  je random seed. Operace (**mod M**) znamená zbytek po celočíselném dělení číslem  $M$ . Číslo  $M$  může mít hodnotu například  $2^{32}$ .



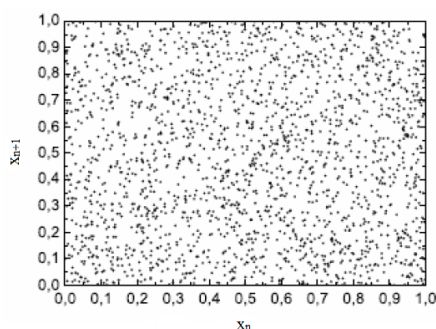


Tyto generátory generují celá čísla s rovnoměrným rozložením z intervalu  $0 \leq x_n \leq M-1$ . Abychom získali hodnotu z intervalu  $0 \leq x_n < 1$  stačí podělit vygenerované číslo číslem  $M$ . Jelikož počet čísel v tomto intervalu je konečný, začnou se generovaná čísla nejpozději po  $M$  generování opakovat (tzv. perioda generátoru) [6].

Problémem těchto generátorů není ani tak periodičita, se kterou se ostatně počítá. Větším problémem je, pokud budeme generovat náhodné body v prostoru a nevhodně zvolíme konstanty  $a$ ,  $c$ ,  $M$ , budou vygenerované body ležet v několika rovnoběžných rovinách, popř. v několika částech prostoru [9]. Nebudou rovnoměrně rozmístěny v prostoru, jak by tomu mělo být (Obrázek 5), ale budou vznikat prázdná místa. Příklady nevhodně zvolených konstant je možné vidět na obrázku 4.



Obrázek 4: Výstupy z kongruentních gen. při nevhodně zvolených konstantách [14]



Obrázek 5: Příklad opravdu náhodně rozložených bodů v prostoru [15]

### 2.3.3 Požadavky na generátory a testování náhodnosti

V této části se budeme bavit o generátorech pseudonáhodných čísel. Jak již bylo výše uvedeno, výsledná posloupnost těchto čísel vychází z deterministického algoritmu a je konečná. Z tohoto důvodu nás budou zajímat míra náhodnosti vygenerovaných čísel a jejich použitelnost.

Požadavky kladené na tyto generátory [13]:

**dlouhá perioda** – velikost generované náhodné posloupnosti by měla být jen zlomkem délky periody generátoru. Perioda by měla být alespoň o řád vyšší než je druhá mocnina počtu generovaných čísel

**nevypočítatelnost** – je důležitou vlastností pro potřeby kryptografie. V reálném čase by z vygenerované posloupnosti nemělo jít určit následující generované číslo.

**opakovatelnost** – měla by existovat možnost generovat opakovaně stejné posloupnosti (možnost volit random seed). Toho se využívá zejména u různých simulačních úloh.

**efektivita a přesnost** – generátor by měl pracovat co nejrychleji a nezabírat systémové prostředky (paměť, procesor). Generátor by měl fungovat shodně na různých platformách a operačních systémech.

Kvalita navržených generátorů se ověřuje pomocí testování náhodných čísel z nich vygenerovaných. Především jsou to [9]:

**empirické testy** – jedná se o statistické metody porovnávající vygenerované posloupnosti s posloupnostmi náhodných čísel. První testy zformuloval D. E. Knuth ve



své knize *The Art of Computer Programming*, nový standard přinesl v roce 1995 George Marsaglia se svým testovacím programem DIEHARD.

**teoretické testy** – jedná se o testy vycházející z teorie čísel.

Tyto testy nám nedávají nikdy úplnou jistotu, že vygenerované posloupnosti čísel jsou skutečně náhodné. Posloupnost čísel vygenerovaných pomocí deterministického algoritmu lze považovat za náhodnou, pokud nebylo testy prokázáno, že se liší od skutečně náhodné posloupnosti čísel.

### 2.3.4 Transformace náhodných čísel na hodnoty náhodných veličin

Generování kvalitních náhodných čísel na intervalu  $(0,1)$ , jak bylo popsáno výše, je jednou z podmínek pro vytvoření simulačního modelu a aplikace metody Monte Carlo. Kromě toho potřebujeme generovat hodnoty náhodných veličin s daným rozdělením (exponenciální, normální atd.).

Mezi nejčastější metody transformace náhodných čísel z intervalu  $(0,1)$  na náhodná čísla s daným rozdělením patří [6], [9]:

- metoda inverzní transformace
- metoda vylučovací (zamítací)
- kompoziční metoda

Pro potřeby této práce podrobněji rozebereme pouze metodu inverzní transformace.

Abychom mohli aplikovat metodu inverzní transformace, je nutné znát distribuční funkci  $F$  náhodné veličiny  $X$ . Mějme náhodnou veličinu  $U$  s rovnoměrným rozdělením na intervalu  $(0,1)$  a necht'  $F(x)$  je libovolná distribuční funkce. Potom náhodná veličina  $X = F^{-1}(U)$  má rozdělení s distribuční funkcí  $F(x)$ . Postup aplikace metody inverzní transformace je takový, že nejdříve se vygeneruje náhodné číslo  $u$  s rovnoměrným rozdělením na intervalu  $(0,1)$ . Toto vygenerované číslo považujeme za hodnotu distribuční funkce v (dosud neznámém) bodě  $x$  (Graf 1). Pak píšeme:

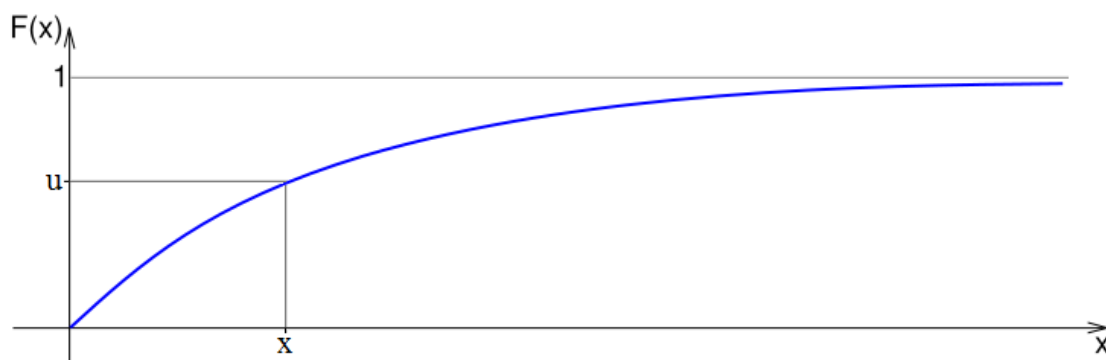
$$F(x) = u \tag{21}$$



Z tohoto vztahu vyjádříme hodnotu  $x$  jako:

$$x = F^{-1}(u), \quad (22)$$

kde  $F^{-1}$  označuje inverzní funkci k distribuční funkci  $F$ .



**Graf 1: Distribuční funkce rozdělení náhodné veličiny [24]**

Metoda inverzní transformace je vhodná pro generování náhodných veličin, u kterých lze vyjádřit inverzní funkci  $F^{-1}$  explicitně (exponenciální, Weibullovo apod.). Čím je tvar inverzní funkce jednodušší, tím snazší a rychlejší je generování.

Tuto metodu můžeme použít i pro případy, kdy nejsme schopni analyticky vyjádřit inverzní funkci  $F^{-1}$ , jako je tomu v případě normálního rozdělení. Pak použijeme nějakou z numerických metod řešení (např. Box-Mullerova transformace, Newtonovy metody) [6], [9].

Konkrétní použití transformace náhodných čísel na hodnoty náhodných veličin pro zvolené programové prostředí Matlab je popsáno v kapitole 3.2.1 a 3.2.2.



## 3 Tvorba simulačního softwaru

### 3.1 Požadavky k vytvoření softwaru

Požadavky, které jsou kladeny na výsledný software:

- Software dokáže simulovat jedno i více kanálové systémy hromadné obsluhy (s paralelním uspořádáním obslužných kanálů).
- Vstupní hodnoty doby do příchodu a odchodu požadavku budou moci být s rozdělením exponenciálním, normálním nebo diskretním.
- Dále bude software umožňovat nastavit počáteční stav a maximální velikost fronty.
- Software bude simulovat jak ustálené, tak i dynamicky se měnící systémy, tj. systémy, u kterých se v čase mění parametry (počet obslužných kanálů, střední doby vstupů a výstupů atd.).
- Bude možné zvolit dobu a počet provedení simulací.
- Jako výsledky simulace dostaneme základní charakteristiky SHO, jako je počet jednotek v systému/frontě, doba strávená v systému/frontě, pravděpodobnost obslužení požadavku bez čekání a pravděpodobnosti jednotlivých stavů.
- Software bude disponovat grafickým uživatelským rozhraním (GUI) pro zadávání vstupů a vyhodnocování výsledků.

Pro vytvoření softwaru simulujícího systémy hromadné obsluhy jsme zvolili programové prostředí Matlab. Konkrétně verzi 7.9.0.529(R2009b) 64-bit(win64).

Matlab jsme zvolili z důvodu toho, že se v něm jednoduše pracuje s maticemi, disponuje grafickým uživatelským rozhraním (GUI), má v sobě implementován kvalitní generátor náhodných čísel a celou řadu matematických a jiných funkcí.

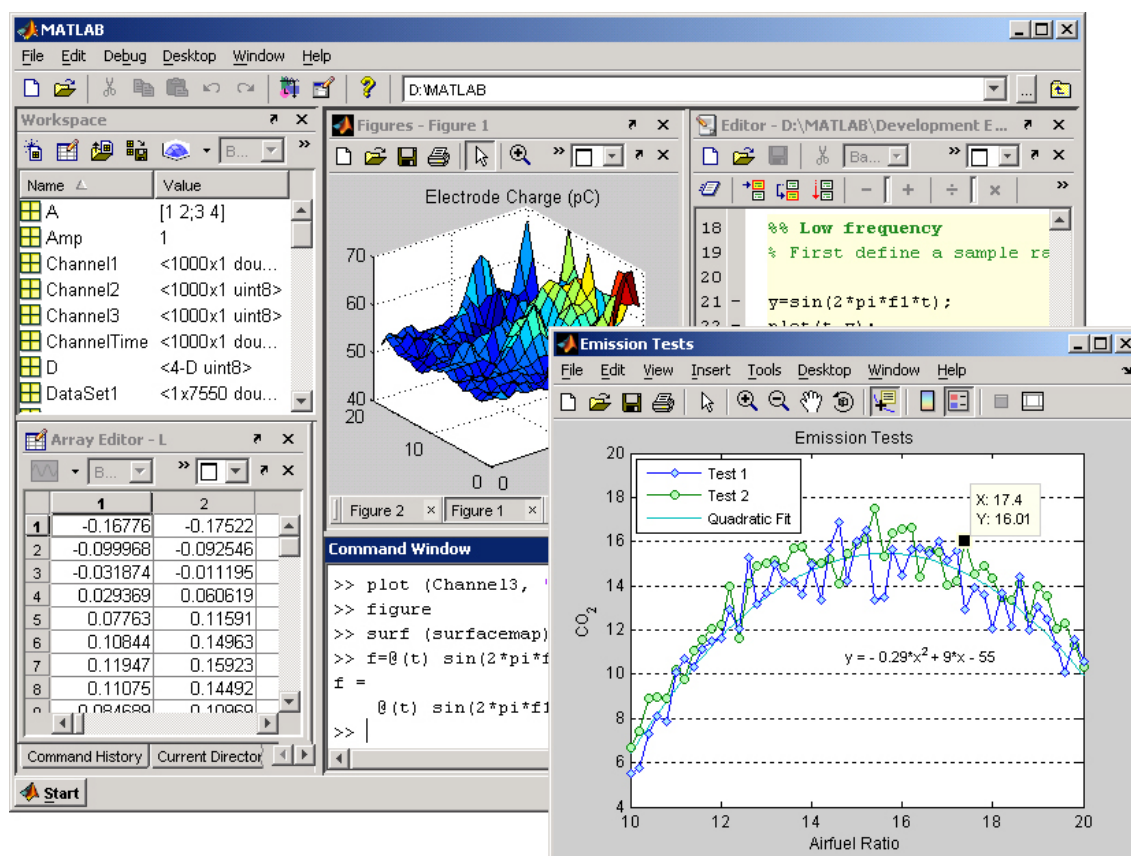
Dalším důvodem bylo, že výsledný software bude sloužit jako učební pomůcka pro studenty Fakulty mechatroniky, informatiky a mezioborových studií na Technické univerzitě v Liberci (předmět Metody užívané v logistice), kteří mají programování a používání Matlabu jako předmět studia.



### 3.2 Programové prostředí Matlab

System Matlab, vyvinutý v roce 1984 firmou *The Mathworks, Inc* v USA, je software, s jehož pomocí lze provádět řadu operací, spojených s matematikou, grafikou, reálným měřením, přenosem dat apod. Pomocí řady rozšiřujících knihoven, tzv. *toolboxů*, se možnosti základního jádra ještě rozšiřují. Uživatel tak může provádět získávání, analýzu, optimalizaci a vizualizaci dat z mnoha technických i jiných oborů [19], [20]. Ukázku programového prostředí Matlab můžeme vidět na obrázku 6.

Název Matlab vznikl zkrácením dvou slov **MATrix** **LABoratory** (volně přeloženo „laboratoř s maticemi“). Jak již název napovídá, tak klíčovou datovou strukturou při výpočtech jsou matice. Vlastní programovací jazyk vychází z jazyka Fortran.



Obrázek 6: Ukázka programového prostředí Matlab [23]

Matlab obsahuje také grafické uživatelské rozhraní nazvané GUIDE (Graphical User Interface Development Environment) [22]. Uživatelské rozhraní můžeme vytvářet



a editovat za pomoci klasických komponent (listbox, tables, button, sliders apod.), které lze za běhu aplikace měnit. Vzhled vytvořené GUI aplikace je ukládán do souboru s příponou \*.fig a vlastní zdrojový kód s příponou \*.m.

### **3.3 Transformace náhodných čísel na hodnoty náhodných veličin v prostředí Matlab**

Matlab disponuje kvalitním generátorem náhodných čísel z intervalu (0,1). Pro potřeby naší aplikace potřebujeme tyto hodnoty transformovat na hodnoty různých rozdělení, konkrétně na exponenciální a normální.

#### **3.3.1 Transformace na exponenciální rozdělení**

Transformaci na hodnoty z exponenciálního rozdělení budeme provádět pomocí výše zmiňované metody inverzní transformace. Ze známé distribuční funkce exponenciálního rozdělení

$$F(t) = 1 - e^{-\lambda t}, \quad (23)$$

můžeme vyjádřit hodnotu  $t$  tak, že za hodnotu distribuční funkce  $F(t)$  dosadíme náhodné číslo  $rand$  z intervalu (0,1). Pak dostaneme:

$$\begin{aligned} rand &= 1 - e^{-\lambda t} \\ e^{-\lambda t} &= 1 - rand = rand \\ t &= -\frac{\ln(rand)}{\lambda} \end{aligned}$$

kde  $rand$  je náhodné číslo z intervalu (0,1),  $\lambda$  je parametr distribuční funkce, v našem případě střední intenzita vstupního respektive výstupního toku a  $t$  je výsledná transformovaná hodnota exponenciálního rozdělení.

V softwarové aplikaci budeme jako parametry vstupu a výstupu zadávat střední dobu do příchodu respektive odchodu požadavku, která je dána vztahem:

$$strDoba = \frac{1}{\lambda} \quad (24)$$



potom výsledný vztah bude vypadat takto:

$$t = -\ln(rand) \cdot strDoba \quad (25)$$

Zdrojový kód zapsaný v Matlabu vypadá následovně:

```
t = -(log(rand()))*strDoba;
```

kde *rand()* je funkce, implementována v Matlabu, která vrací náhodnou hodnotu z intervalu (0,1).

### 3.3.2 Transformace na normální rozdělení

Protože nejsme schopni analyticky vyjádřit distribuční funkci normálního rozdělení, použijeme k transformaci na hodnoty z normálního rozdělení tzv. Box-Mullerovu transformaci.

Nejprve normální rozdělení s parametry  $N(\mu, \sigma^2)$  převedeme na normované normální rozdělení s parametry  $N(0, 1)$ , pak má náhodná veličina tvar:

$$u = \frac{t - \mu}{\sigma} \quad (26)$$

Pomocí Box-Mullerovy transformace [21] můžeme ze dvou nezávislých náhodných veličin s rovnoměrným rozdělením na intervalu (0,1) vytvořit dvě nezávislé náhodné veličiny s normovaným normálním rozdělením.

$$\begin{aligned} u_1 &= \sqrt{-2\ln \cdot rand_1} \cdot \cos(2\pi \cdot rand_2) \\ u_2 &= \sqrt{-2\ln \cdot rand_1} \cdot \sin(2\pi \cdot rand_2) \end{aligned} \quad (27)$$

Potom můžeme hodnotu  $t$  vyjádřit jako:

$$t = u_1 \cdot \sigma + \mu, \quad (28)$$

kde  $rand_1$  a  $rand_2$  jsou náhodná čísla z intervalu (0,1),  $u_1$  je hodnota s normovaným normálním rozdělením (je jedno, kterou zvolíme, jestli  $u_1$  nebo  $u_2$ ),  $\sigma$  představuje směrodatnou odchylku vstupního respektive výstupního toku,  $\mu$  je střední doba do příchodu respektive odchodu požadavku a  $t$  je výsledná transformovaná hodnota normálního rozdělení.





Algoritmus zapsaný v Matlabu vypadá takto:

```
t = (sqrt(-2*log(rand()))*cos(2*pi*rand()))*smerOdch + strDoba;
```

### 3.4 Princip simulace

Simulace v programu Matlab je tvořena pomocí metody Monte Carlo, kdy se mnohonásobně krát opakuje stejný náhodný pokus, ale s odlišnými podmínkami pokusu. Odlišné podmínky pokusu jsou dány časově náhodnými vstupy/výstupy ze systému.

Průběh simulace lze rozdělit do následujících kroků:

#### krok 1

Nejprve si do matice nazvané *Generátor* vygenerujeme náhodné okamžiky vstupů požadavků do systému na základě zvoleného rozdělení. Ke každému z nich si vygenerujeme náhodnou dobu jeho obsluhy, dle zvoleného rozdělení. Na základě zadáných parametrů ke každému vstupnímu požadavku vypočteme okamžik výstupu ze systému. Následně hodnoty v matici *Generátor* zaokrouhlíme na celá čísla.

Okamžiky vstupů, výstupů a dob strávených v systému budeme pro lepší představu uvádět v minutách. Tabulka 1 představuje příklad matice *Generátor*, kde je zobrazeno prvních pět požadavků s tím, že první požadavek přišel v čase 10 minut.

Tabulka 1: Ukázka matice *Generátor*

číslo požadavku	okamžik vstupu (min)	doba obsluhy (min)	okamžik výstupu (min)
1	10	2	12
2	15	4	19
3	17	3	22
4	18	4	26
5	28	2	30
...	...	...	...

#### krok 2

Dále budeme pracovat s maticí *Realizace*, která má rozměry počtu realizací krát velikost intervalu simulace. Této matici před začátkem simulace přiřadíme nulové hodnoty.



Vezmeme první požadavek, zjistíme jeho okamžik příchodu do systému, a od následující hodnoty přičítáme jedničku do prvního řádku matice *Realizace* až do okamžiku výstupu požadavku ze systému.

Tento postup provedeme pro každý příchozí požadavek. Pro další realizaci opakujeme krok 1 a v matici *Realizace* se posouváme o řádek níž. Příklad matice *Realizace* po první realizaci představuje tabulka 2.

Tabulka 2: Ukázka matice *Realizace*

číslo realizace	interval simulace (min)																							
	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	...		
1	0	1	1	0	0	0	1	1	2	3	2	2	2	1	1	1	1	0	0	1	1	...		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...		
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...		
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...		
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...		
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...		

### krok 3

V matici *Realizace* máme tedy pro každou minutu stav, ve kterém se systém vyskytoval v každé z náhodných realizací.

Do matice *Stav* si spočítáme četnost výskytu jednotlivých stavů pro každou minutu. Potom matici *Stav* podělíme počtem realizací simulace a dostáváme pravděpodobnost výskytu jednotlivých stavů v každé minutě. Ukázku matice *Stav* představuje tabulka 3.

Tabulka 3: Ukázka matice *Stav*

číslo stavu	interval simulace (min)																	
	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...	
0	0,26	0,22	0,21	0,21	0,2	0,22	0,22	0,23	0,21	0,27	0,24	0,22	0,18	0,20	0,24	0,24	...	
1	0,20	0,26	0,2	0,24	0,19	0,19	0,21	0,20	0,20	0,14	0,18	0,13	0,20	0,20	0,24	0,24	...	
2	0,16	0,15	0,18	0,15	0,16	0,16	0,14	0,14	0,13	0,09	0,11	0,16	0,19	0,19	0,12	0,14	...	
3	0,09	0,08	0,09	0,12	0,16	0,12	0,09	0,11	0,10	0,15	0,10	0,12	0,12	0,13	0,16	0,10	...	
4	0,12	0,12	0,15	0,08	0,07	0,09	0,07	0,07	0,10	0,11	0,13	0,15	0,11	0,11	0,07	0,10	...	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

### krok 4

Ze známých pravděpodobností jednotlivých stavů vypočítáme charakteristiky systému v daný časový okamžik nebo jejich střední hodnoty.



## 4 Způsob použití softwaru SHO-MMC Simulator

Výsledný software, který simuluje úlohy z teorie hromadné obsluhy pomocí metody Monte Carlo, jsme nazvali SHO-MMC Simulator. Software splňuje všechny požadavky, které byly kladeny v kapitole 3.

SHO-MMC Simulator je vytvořen v programovacím prostředí Matlab ve verzi 7.9.0.529, 64-bit. Disponuje grafickým uživatelským rozhraním pro zadávání vstupů a vyhodnocování výsledků.

Software je určen jako výuková pomůcka studentům, kteří mají zapsaný předmět MUL (Metody užívané v logistice) na Fakultě mechatroniky, informatiky a mezioborových studií na Technické univerzitě v Liberci.

### 4.1 Způsob nainstalování a spuštění softwaru

Software SHO-MMC Simulator se skládá ze dvou souborů. Prvním z nich je *SHO\_MMC\_Simulator.fig*, ve kterém je uloženo grafické uživatelské rozhraní s komponentami. Vlastní zdrojový kód je potom uložen v souboru *SHO\_MMC\_Simulator.m*. Software je zkompilován také do podoby samostatně spustitelného souboru *SHO\_MMC\_Simulator.exe*.

Ke spuštění softwaru SHO-MMC Simulator je potřeba mít nainstalovaný Matlab Compiler Runtime (MCR) v příslušné verzi, což je soubor sdílených knihoven, které umožňují spouštění zdrojových kódů Matlabu (M-files). Instalační soubor MCR (MCRInstaller.exe) je přiložen k výslednému softwaru.

Pokud máme nainstalované programové prostředí Matlab, můžeme software spouštět pomocí zdrojového kódu přímo v Matlabu.



**SHO\_MMC\_Simulator**  
Program Realizace Nápoředa

**Nastavení parametrů intervalů**

Typ rozdělení vstupu  
☐ diskretní ☒ exponenciální ☐ normální

střední doba vstupu (t) 3

Typ rozdělení výstupu  
☐ diskretní ☒ exponenciální ☐ normální

střední doba výstupu (t) 2

**Tabulka intervalů s parametry**

	typ vstupu (1-ds 2-ex 3-no)	perioda/ stř. doba vstupu (no rozd.)	sm. odch. vstupu (no rozd.)	typ výstupu (1-ds 2-ex 3-no)	perioda/ stř. doba výstupu (no rozd.)	sm. odch. výstupu (no rozd.)	počet obsaz. kanálů	počet pož. ve frontě	max. vel. fronty	interval OD	interval DO
1	2	3	1	2	2.5000	1	1	0	0	100	1000
2	2	3	1	2	2	1	1	0	0	100	2000

Vložit interval Smažit interval Smažit vše

**Výhodnocení a zobrazení grafů**

interval vyhodnocení Od (t) 1 interval vyhodnocení Do (t) 2000 číslo stavu 6 max. stav 34

střední hodnoty :

Počet požadavků v systému (t)	3.3702
Doba strávená v systému (t)	10.1107
Počet požadavků ve frontě (t)	2.6223
Doba strávená ve frontě (t)	7.867

Číslo stavu střední hodnota pravděpodobnosti

1	0	0.2521
2	2	0.1332
3	5	0.0552
4	6	0.0425

Prs. v (t) Sř. hod. prs. Součet prs.

Prs. obslužená požadavku (t) 0.2521 Sř. hod. prs. 0.48295

**Realizace**

počet realizací 100

**Realizuj**

Obrázek 7: Software SHO-MMC Simulator



## 4.2 Popis jednotlivých funkčních panelů softwaru

Na obrázku 7 je zobrazená výsledná podoba simulačního softwaru SHO-MMC Simulator a jeho rozdělení na jednotlivé funkční panely:

- Nastavení parametrů intervalů,
- Tabulka intervalů s parametry,
- Realizace,
- Vyhodnocení a zobrazení grafů.

V dalších kapitolách budou popsány jednotlivé funkční panely, jejich funkce a použití.

### 4.2.1 Nastavení parametrů intervalů

Na tomto panelu, umístěném vlevo nahoře, můžeme nastavovat parametry jednotlivých intervalů. Počet těchto intervalů může být více, kdy se předpokládá, že každý má odlišné vstupní/výstupní parametry. Pro všechny zadávané číselné hodnoty se používá jako desetinný oddělovač tečka (14.8; 3.1).

#### *Typ rozdělení vstupu*

Zde se zadává typ statistického rozdělení, s jakým budou požadavky vstupovat do systému. Máme na výběr ze tří možností diskrétní, exponenciální a normální rozdělení.

**diskrétní rozdělení** – u diskrétního vstupu zadáváme časovou periodu, se kterou budou požadavky vstupovat do systému

Typ rozdělení vstupu

☒ diskrétní ☐ exponenciální ☐ normální

perioda vstupu (t) 2.3

**exponenciální rozdělení** – zde se zadává střední doba, se kterou přicházejí požadavku do systému

Typ rozdělení vstupu

☐ diskrétní ☒ exponenciální ☐ normální

střední doba vstupu (t) 5.5



**normální rozdělení** – u normálního rozdělení se zadává střední doba, se kterou přicházejí požadavky do systému a navíc její směrodatná odchylka.

Typ rozdělení vstupu

☐ diskrétní ☐ exponenciální ☒ normální

střední doba vstupu (t) 7

směrodatná odchylka vstupu (t) 3

### **Typ rozdělení výstupu**

Zde volíme typ rozdělení, s jakým budou požadavky obsluhovány neboli s jakým budou vystupovat ze systému. Opět máme na výběr ze tří možných rozdělení diskrétní, exponenciální a normální. Způsob zadávání hodnot je obdobný jako u výše popsaného typu rozdělení vstupů.

Výše popsané zadávané hodnoty vstupů a výstupů jsou omezeny a nesmí být menší nežli 0.5. Toto omezení je uděláno z důvodu zabránění generování příliš velkého množství hodnot v samotné simulaci a z toho vyplývající výpočetní náročnost a prodloužení doby simulace. Omezení lze obejít změnou časových jednotek  $hod \Rightarrow min \Rightarrow s$ .

Omezení se netýká směrodatných odchylek, které musí být kladné (v případě zvolení 0, přechází normální rozdělení na diskrétní).

Dále budeme zadávat hodnoty týkající se obslužných kanálů, počátečních podmínek simulace a parametrů fronty.

**počet obslužných kanálů** – zde se zadává počet paralelně řazených obslužných kanálů, kterými bude simulovaný systém disponovat. Systém musí obsahovat minimálně jeden obslužný kanál.

počet obslužných kanálů 4

Následuje zadání dvou parametrů,

**počet obsazených kanálů**

počet obsazených kanálů

0

**počet požadavků ve frontě**

počet požadavků ve frontě

0

kteřé představují počáteční podmínky neboli počáteční stav, ve kterém se systém nachází na začátku simulace nebo na začátku každého vloženého intervalu. Počáteční stav získáme tak, že sečteme počet obsazených kanálů a počet požadavků čekajících ve frontě. Je zřejmé, že počet obsazených kanálů nemůže být větší než počet obslužených kanálů.

**maximální velikost fronty** – nám udává maximální počet požadavků čekajících ve frontě. Požadavky nad velikost fronty jsou systémem odmítnuty bez obslužení. Když k tomuto číslu přičteme počet obslužených kanálů, dostaneme maximální počet požadavků v systému. U simulací popsaných pomocí Kendallový klasifikace s neomezenou délkou fronty, je vhodné zadat vysokou hodnotu dle uvážení.

maximální velikost fronty

50

Poslední dva parametry nám budou specifikovat dobu trvání simulace.

**interval Od (t)**

interval Od (t)

1

**interval Do (t)**

interval Do (t)

1000

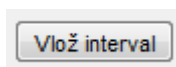
Tyto zadávané hodnoty specifikují interval, pro který platí výše zvolené parametry. Čím větší zvolíme interval, tím přesnější dostaneme výsledky, ale také zvýšíme výpočetní náročnost simulace. První interval vždy začíná od 1.



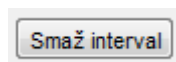
#### 4.2.2 Tabulka intervalů s parametry

V tabulce intervalů, která je umístěna vpravo nahoře, jsou zobrazeny jednotlivé intervaly se zvolenými parametry, které jsme zadávali v předchozí kapitole. Každý řádek představuje interval a ve sloupcích jsou specifikovány jednotlivé parametry intervalu. Na panelu jsou tlačítka pro vkládání a mazání intervalů.

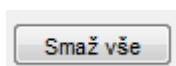
**Vlož interval** – vkládá intervaly se zvolenými parametry do tabulky intervalů. Každý z těchto intervalů může mít odlišné vstupní/výstupní parametry.



**Smaž interval** – maže po jednom intervaly od naposledy vloženého.



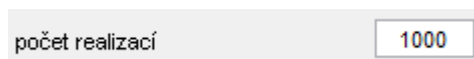
**Smaž vše** – smaže všechny vložené intervaly.



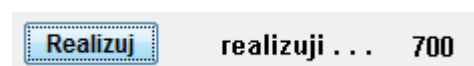
#### 4.2.3 Realizace

Na tomto panelu, který se nachází vlevo dole, je umístěno tlačítko pro spuštění realizace simulace. Dále zde nastavujeme počet realizací simulace. Vedle tlačítka je zobrazována informace o aktuálním průběhu realizace.

**počet realizací** – tento parametr označuje, kolikrát bude provedena realizace simulace metodou Monte Carlo. S větším počtem realizací stoupá přesnost výsledných hodnot ( $\frac{1}{\sqrt{n}}$ ). Zároveň ale stoupá lineárně výpočetní a časová náročnost simulace  $O(n)$



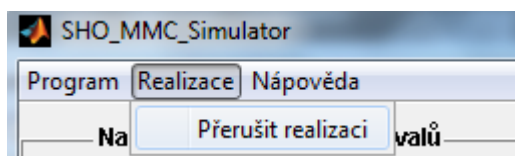
**Realizuj** – realizuje simulaci podle nastavených parametrů.







Spuštěnou realizaci simulace lze přerušit stiskem tlačítka Přerušit realizaci, které se nachází nahoře v hlavním menu pod nabídkou Realizace. Tlačítko je aktivní pouze v průběhu realizace.



Nasimulované hodnoty ukládáme do matice, která má rozměry počtu realizací krát velikost intervalu (u dynamických systémů celkového intervalu). Jelikož máme k dispozici omezené množství paměti, není vhodné zadávat příliš velké intervaly (např. 100 000) při velkém počtu realizací (např. 1 000 000), kde by se počet hodnot v matici vyšplhal na  $10^{11}$ . Možné maximální zadávané hodnoty počtu realizací a celkového intervalu se odvíjí od konkrétního výpočetního stroje, na kterém budeme provádět simulace. Na běžných počítačích by počet hodnot v matici neměl překročit  $10^8$ .

#### 4.2.4 Vyhodnocení a zobrazení grafů

Panel Vyhodnocení a zobrazení grafů se nachází vpravo pod Tabulkou intervalů s parametry. Na tomto panelu se vyhodnocují a zobrazují hodnoty získané simulací.

**interval vyhodnocení Od/Do (t)** – tyto parametry specifikují interval, ze kterého se bude provádět vyhodnocení a výpočty charakteristik zkoumaného systému.

interval vyhodnocení Od (t)	interval vyhodnocení Do (t)
<input type="text" value="1"/>	<input type="text" value="1000"/>

Pod intervaly vyhodnocení se nacházejí tlačítka pro zobrazení základních charakteristik systémů hromadné obsluhy (viz kapitola 1.3).

**Počet požadavků v systému/frontě (t)** – tlačítka, která spočtou a zobrazí střední hodnoty počtu požadavků v systému/frontě v zadaném intervalu vyhodnocení. Dále vykreslí grafy průběhu okamžitého a průměrného počtu požadavků v systému/frontě v zadaném intervalu vyhodnocení.



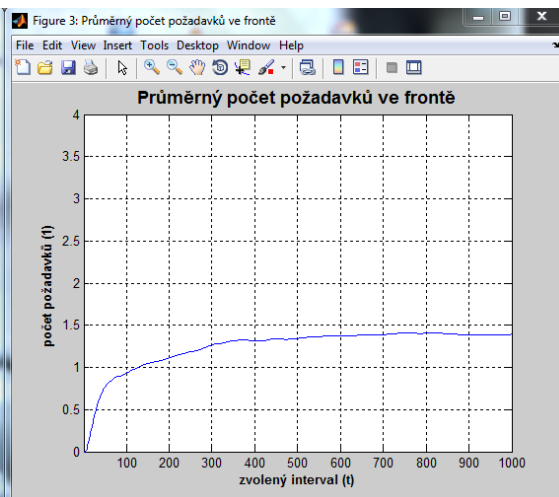
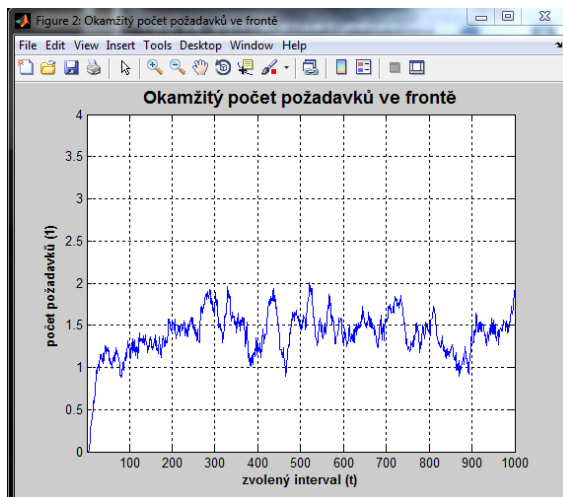
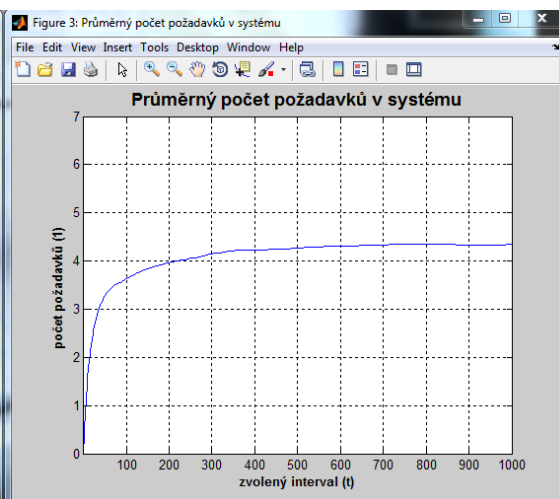
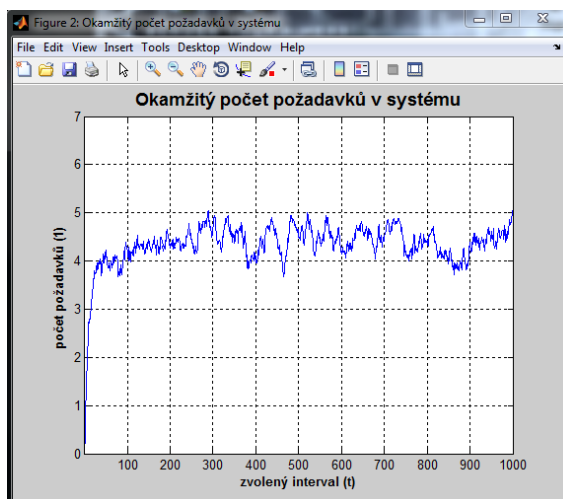
Výsledné hodnoty statických a dynamických systémů se vypočítávají podle vztahu pro  $n_s$  (2) a  $n_f$  (4) pro jednonálové SHO, respektive podle vztahu pro  $n_s$  (13) a  $n_f$  (15) pro vícekanálové SHO.

Počet požadavků v systému (t)

4.3454

Počet požadavků ve frontě (t)

1.3932

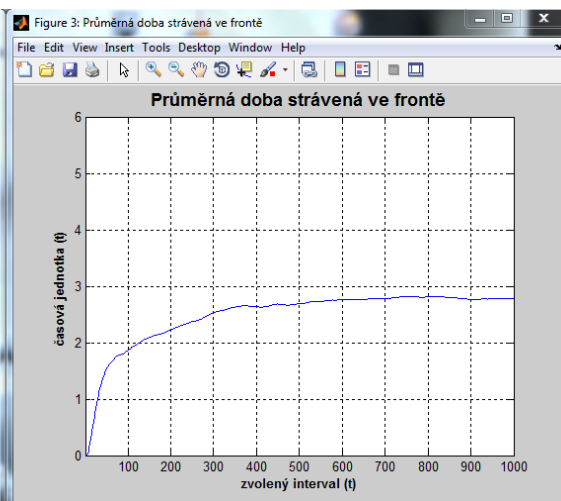
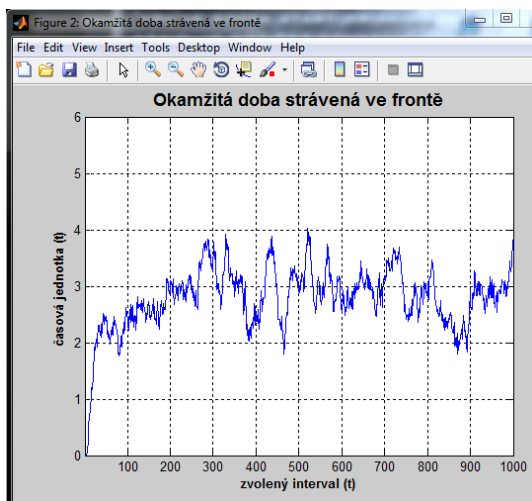
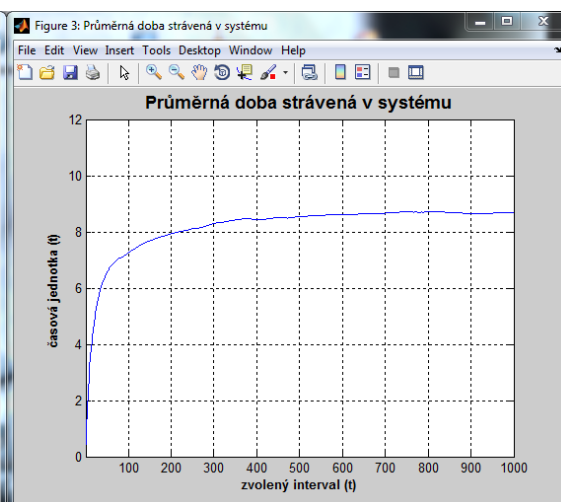
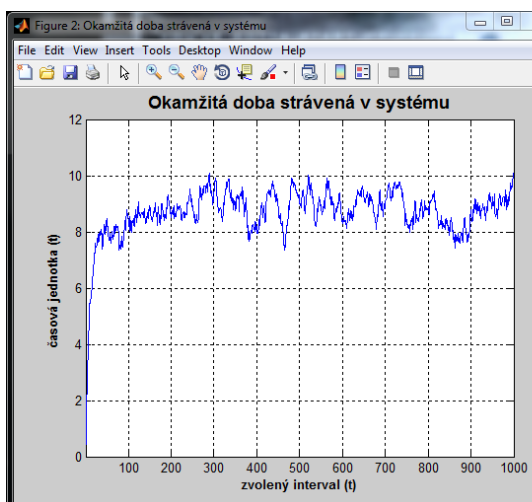


**Doba strávená v systému/frontě (t)** – tlačítka, která spočtou a zobrazí střední hodnoty časových jednotek strávených v systému/frontě v zadaném intervalu vyhodnocení. Dále vykreslí grafy průběhu okamžitého a průměrného počtu časových jednotek strávených v systému/frontě v zadaném intervalu vyhodnocení.

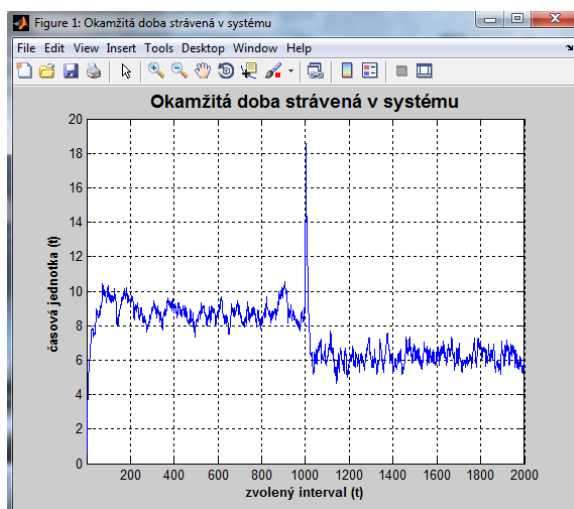


Výsledné hodnoty statických systémů se vypočítávají podle vztahu pro  $t_s$  (6) a  $t_f$  (8) pro jednocanálové SHO, respektive podle vztahu pro  $t_s$  (17) a  $t_f$  (19) pro vícekanálové SHO.

Doba strávená v systému (t)	8.6908
Doba strávená ve frontě (t)	2.7864



U dynamických systémů, u kterých se v průběhu realizace simulace mění střední doby vstupu, se výsledné hodnoty nezobrazovaly korektně (časově nespojitá hodnota středních dob v systému/frontě). V grafech průběhu okamžitého počtu časových jednotek vznikaly velké skoky, než se systém ustálil (Graf 2).



Graf 2: Nespojitosť v grafu doby strávené v systéme u dynamického SHO

Z tohoto dôvodu byl pro výpočet približných hodnot doby strávené v systéme/frontě pro jednocanálové i vícekanálové SHO použit vzťah [16], [17]:

$$t_{s/f} = \frac{\sum_{n=1}^{\infty} p_n \cdot \left\lfloor \frac{n+x-1}{x} \right\rfloor \cdot \frac{1}{\mu}}{1-p_0}, \quad (29)$$

kde  $n$  je číslo stavu,  $p_n$  pravdepodobnosť stavu,  $x$  je počet obslužných kanálov v systéme a  $\mu$  je stredná intenzita výstupného toku. Znaky  $\lfloor \quad \rfloor$  predstavujú dolnú celou časť čísla.

**Prs. obslužení požadavku (t)** – tlačítko, ktoré spočte a zobrazí strednú hodnotu pravdepodobnosti v zadanom intervale vyhodnocení, že bude príchozí požiadavek bez čakania obslužen. Ďalej vykreslí grafy prubehu okamžite a průměrné pravdepodobnosti v zadanom intervale vyhodnocení, že bude príchozí požiadavek bez čakania obslužen. Grafy vypadajú obdobne, ako je tomu u predchádzajúcich charakteristik systému (Počet požadavkov v systéme/frontě).

Prs. obslužení požadavku (t) 0.50177

V pravej polovine panelu Vyhodnocení a zobrazení grafů se nachází tabulka stavů, které po zvolení chceme vyhodnocovat a zobrazovat. V pravém horním rohu



panelu se zobrazuje informace o počtu stavů, ve kterých se systém v průběhu simulace vyskytoval (max. stav).

**číslo stavu** – zde se zadává číslo stavu, který chceme vyhodnocovat a zobrazovat.

číslo stavu

**Vlož stav/vše** – tlačítka pro vkládání zvolených stavů/všech stavů, ve kterých se systém v průběhu realizace vyskytoval, do tabulky stavů.

Vlož stav Vlož vše

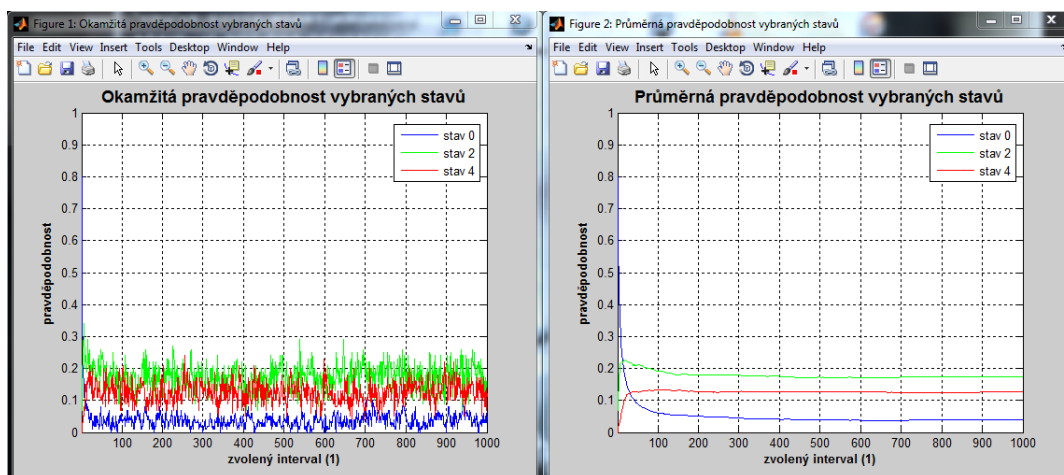
**Smaž stav/vše** – tlačítka pro mazání zvolených stavů/všech stavů najednou z tabulky stavů.

Smaž stav Smaž vše

Pod tabulkou stavů se nachází tlačítka pro vyhodnocování zvolených stavů.

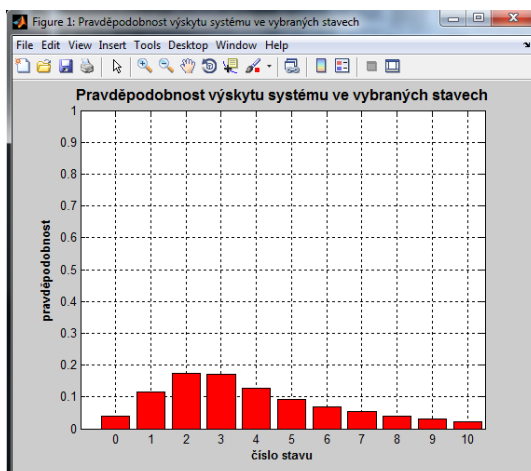
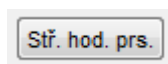
**Prs. v (t)** – tlačítko, které nám vykreslí grafy průběhu okamžité a průměrné pravděpodobnosti zvolených stavů v zadaném intervalu vyhodnocení. Počet stavů, které lze najednou vykreslit je omezen na 10.

Prs. v (t)





**Stř. hod. prs.** – tlačítko, které vykreslí graf středních hodnot pravděpodobnosti zvolených stavů v zadaném intervalu vyhodnocení. Dále doplní do tabulky stavů k jednotlivým stavům jejich střední hodnoty pravděpodobnosti.



**Součet prs.** – tlačítko, které provede součet středních hodnot pravděpodobnosti zvolených stavů v zadaném intervalu vyhodnocení a tuto hodnotu zobrazí.



### 4.3 Postup simulace

Nejprve na panelu Nastavení parametrů intervalů (4.2.1) zvolíme námi požadované parametry a interval (dobu), pro který budou tyto parametry platit a vložíme je do Tabulky intervalů (4.2.2). Pokud simulujeme systém, u kterého se budou zvolené parametry v čase měnit, změníme parametry a vložíme je do Tabulky intervalů jako další interval. Při zadávání všech číselných parametrů použijeme jako desetinný oddělovač tečku.

Po vložení všech intervalů s parametry nastavíme na panelu Realizace (4.2.3) počet realizací a stiskneme tlačítko Realizuj. Vyčkáme čas, než proběhne celá simulace a přejdeme k panelu Vyhodnocení a zobrazení grafů (4.2.4).

Nejprve nastavíme interval vyhodnocení, tj. interval, ze kterého se budou určovat výsledky (implicitně nastaven na celou délku simulace). Klikneme na námi



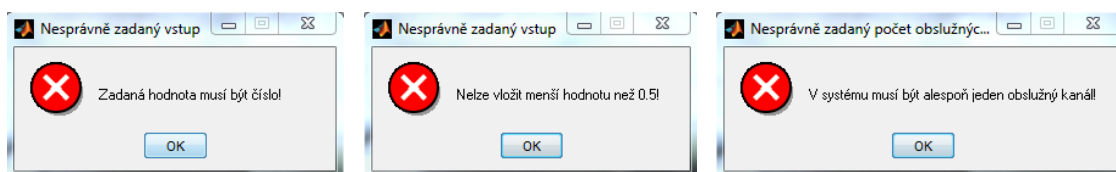
požadované tlačítko pro výsledek. Vyčkáme, než software vykreslí grafy a zobrazí střední hodnotu.

Pro vyhodnocení jednotlivých stavů zvolíme stavy, u kterých chceme zjišťovat výsledky, a vložíme je do Tabulky stavů. Pod tabulkou klikneme na námi požadované tlačítko pro výsledek.

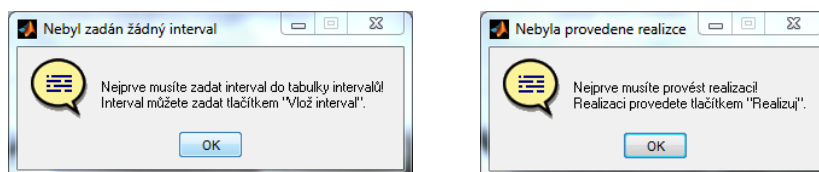
#### 4.4 Ošetření výjimek

Výjimkami rozumíme neočekávané situace (chyby), ke kterým může dojít při běhu softwaru. Například nevhodně zadaný uživatelský vstup (zadaná nečíselná hodnota) nebo spuštění realizace simulace, bez toho aniž bychom zadali vstupní parametry.

Pokud zadáme špatný vstup, objeví se dialogové okno s popisem chyby a doporučením správného vstupu.



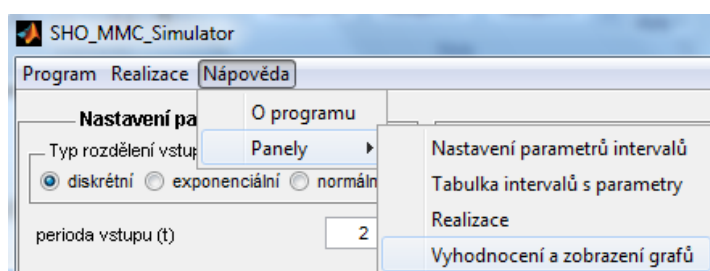
Další výjimky se týkají například špatného postupu při práci se softwarem. U těchto výjimek se objeví informativní okno s popisem a doporučením jak postupovat.



#### 4.5 Náповěda

K softwaru SHO-MMC Simulator je vytvořena jednoduchá nápověda, popisující postup práce se softwarem. Dále jsou v ní popsány jednotlivé funkční panely softwaru s komponenty.

Nápovědu k softwaru nalezneme nahoře v hlavním menu pod nabídkou Nápověda.







## 5 Příklady použití softwaru

Simulace budeme provádět na notebooku DELL Studio 1555 s procesorem Pentium Dual-Core T4200 (2,00 GHz), 4 GB RAM, Windows 7 (64-bit).

### 5.1 Statické SHO

#### 5.1.1 Jednakanálový SHO

Budeme simulovat stánek rychlého občerstvení. Víme, že ke stánku přijde v průměru 20 zákazníků za hodinu. Obsluha jednoho zákazníka trvá průměrně 2 minuty. Ve stánku pracuje jeden prodavač.

Předpokládáme, že příchody zákazníků i jejich obsluha jsou veličiny s exponenciálním rozdělením. Dále předpokládáme neomezeně dlouhou frontu typu FIFO. Pomocí Kendallovy klasifikace bychom systém popsali následovně:

$$M/M/1/\infty/FIFO$$

Nejprve si ze zadaných hodnot vyjádříme střední doby vstupu a střední doby obsluhy (výstupu) požadavků (zákazníků).

$$\text{stř. doba vstupu} = \frac{60}{20} = 3 \text{ min}$$

$$\text{stř. doba výstupu} = 2 \text{ min}$$

Nyní můžeme začít zadávat hodnoty do softwaru, zvolíme exponenciální rozdělení vstupů a výstupů. Jako střední dobu vstupu zadáme hodnotu 3, jako střední dobu výstupu hodnotu 2. Počet obslužných kanálů bude 1. Počáteční podmínky předpokládáme nulové, proto jako hodnoty počtu obsazených kanálů a počtu požadavků ve frontě zadáme 0. Maximální velikost fronty předpokládáme neomezenou, proto zvolíme vysoké číslo, např. 1000. Pravděpodobnost, že okamžitá délka fronty v průběhu simulace překročí zvolenou hodnotu je velmi malá (takřka nulová). Zadané hodnoty do softwaru ukazuje následující screenshot ze softwaru:



**Nastavení parametrů intervalů**

Typ rozdělení vstupu  
☐ diskrétní ☒ exponenciální ☐ normální

střední doba vstupu (t)

Typ rozdělení výstupu  
☐ diskrétní ☒ exponenciální ☐ normální

střední doba výstupu (t)

počet obslužených kanálů

počet obsazených kanálů

počet požadavků ve frontě

maximální velikost fronty

interval Od (t)

interval Do (t)

**Realizace**

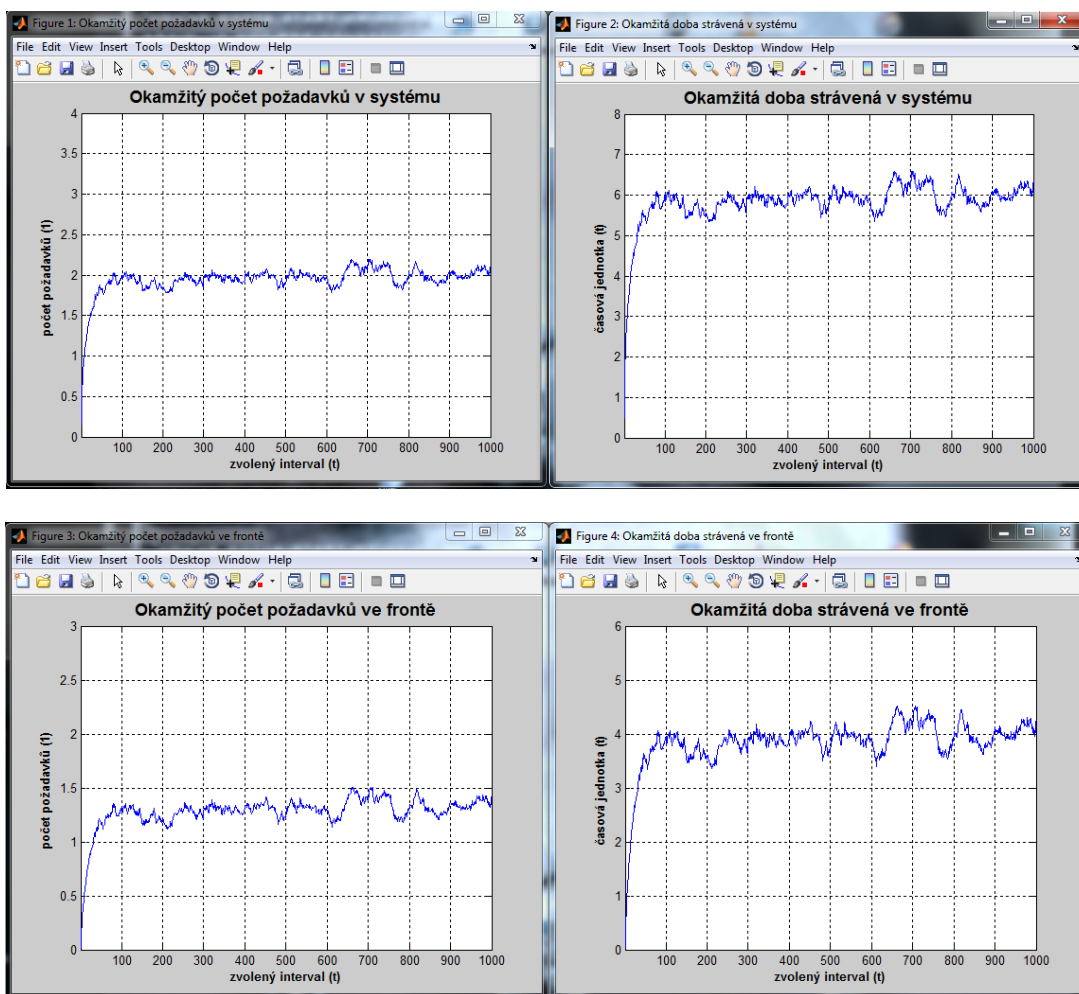
počet realizací

**Realizuj**

Budeme provádět několik simulací a ukážeme si, jak přesnosti výsledků ovlivňuje délka intervalu realizace simulace a počet realizací simulace (tabulka 4). V tabulce 4 můžeme porovnat charakteristiky zkoumaného systému (stánku rychlého občerstvení) získané simulací (SIM) s hodnotami získanými analyticky (ANL). Dále je v tabulce zobrazena časová náročnost simulace pro zvolené parametry simulace.

**Tabulka 4: Charakteristiky statického jednobanového SHO**

interval Od (min)	interval Do (min)	počet realizací	stř. počet požadavků v systému (1)		stř. doba strávená v systému (min)		stř. počet požadavků ve frontě (1)		stř. doba strávená ve frontě (min)		časová náročnost simulace (s)
			SIM	ANL	SIM	ANL	SIM	ANL	SIM	ANL	
1	100	10	1,5320	2	4,5960	6	0,8920	1,3333	2,6760	4	0,1
1	100	100	1,6459	2	4,9377	6	1,0235	1,3333	3,0705	4	0,5
1	100	1000	1,6209	2	4,8626	6	1,0003	1,3333	3,0008	4	1,5
1	1000	10	2,1076	2	6,3228	6	1,4198	1,3333	4,2594	4	0,5
1	1000	100	1,9739	2	5,9217	6	1,3069	1,3333	3,9207	4	1,5
1	1000	1000	1,9704	2	5,9113	6	1,3065	1,3333	3,9196	4	12
1	1000	10000	1,9708	2	5,9124	6	1,3079	1,3333	3,9238	4	116
1	10000	10	2,0915	2	6,2746	6	1,4182	1,3333	4,2545	4	1,6
1	10000	100	1,9997	2	5,9992	6	1,3346	1,3333	4,0037	4	12,5
1	10000	1000	2,0039	2	6,0117	6	1,3373	1,3333	4,012	4	119

**Graf 3: Charakteristiky statického jednonálového SHO**

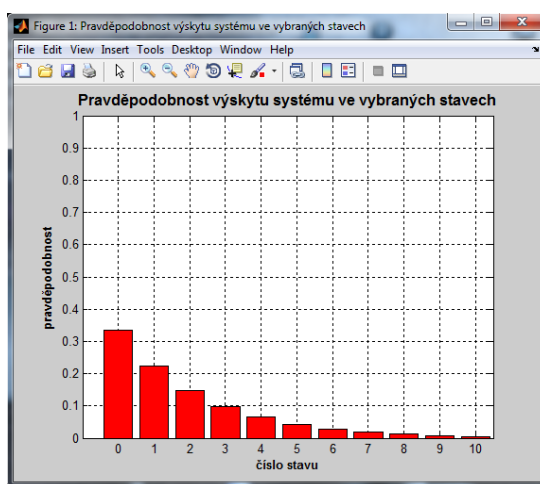
Z výsledných grafů charakteristik zkoumaného systému (Graf 3) vyplývá, že se systém ustaluje až přibližně po 100 minutách. Proto výsledky uvedené v tabulce 4, jsou pro zvolený interval 1-100 minut velmi nepřesné (v porovnání s intervalem 1 až 1000) a počet realizací nemá na tyto výsledky příliš vliv. S prodloužením intervalu (pro systém, který se ustálí) a větším počtem realizací stoupá i přesnost výsledných ustálených charakteristik.

V tabulce 5 můžeme porovnat střední hodnoty pravděpodobností jednotlivých stavů získaných pomocí simulace s hodnotami získanými analyticky. V tabulce je zobrazeno prvních 11 stavů systému. Střední hodnoty pravděpodobností stavů jsou zobrazeny v grafu 4.



Tabulka 5: Pravděpodobnosti stavů statického jednokanálového SHO

interval Od (min)	interval Do (min)	počet realizací	Pravděpodobnosti stavů											
			číslo stavu											
			0	1	2	3	4	5	6	7	8	9	10	
analytické hodnoty			0,3333	0,2222	0,1481	0,0988	0,0658	0,0439	0,0292	0,0195	0,0130	0,0087	0,0058	
1	100	10	0,3740	0,2520	0,1420	0,089	0,0490	0,0120	0,0130	0,0210	0,0120	0,0120	0,0100	
1	100	100	0,3731	0,2202	0,1433	0,1026	0,0605	0,0418	0,0239	0,0130	0,0091	0,0062	0,0035	
1	100	1000	0,3785	0,2393	0,1435	0,0899	0,0558	0,0351	0,0216	0,0145	0,0080	0,0048	0,0030	
1	1000	10	0,3387	0,2257	0,1375	0,1097	0,0750	0,0444	0,0273	0,0174	0,0105	0,0058	0,0038	
1	1000	100	0,3331	0,2256	0,1466	0,1006	0,0646	0,0407	0,0280	0,0204	0,0132	0,0097	0,0064	
1	1000	1000	0,3361	0,2240	0,1479	0,0982	0,0654	0,0430	0,0288	0,0192	0,0127	0,0084	0,0056	
1	1000	10000	0,3367	0,2231	0,1482	0,0981	0,0651	0,0431	0,0288	0,0191	0,0127	0,0085	0,0057	
1	10000	10	0,3320	0,2237	0,1499	0,0974	0,0644	0,0445	0,0305	0,0191	0,0130	0,0093	0,0065	
1	10000	100	0,3346	0,2227	0,1487	0,0976	0,0654	0,0434	0,0291	0,0199	0,0131	0,0085	0,0057	
1	10000	1000	0,3327	0,2219	0,1479	0,0988	0,0659	0,0440	0,0293	0,0196	0,0131	0,0088	0,0058	



Graf 4: Pravděpodobnosti stavů statického jednokanálového SHO

Dále si ukážeme, jak změna pravděpodobnostního rozdělení ovlivní výsledné charakteristiky systému.

Použijeme stejný příklad se stánkem rychlého občerstvení. Opět víme, že ke stánku přijde v průměru 20 zákazníků za hodinu. Ale budeme uvažovat, že obsluha jednoho zákazníka trvá v průměru 2 minuty  $\pm$  15 sekund.

Předpokládáme, že příchod zákazníků je veličina s exponenciálním rozdělením a jejich obsluha veličina s normálním rozdělením. Vše ostatní předpokládáme stejné, jako u předchozího příkladu. Kendallovou klasifikací bychom systém popsali následovně:

M/N/1/ $\infty$ /FIFO



Nejprve si ze zadaných hodnot vyjádříme střední doby vstupu a střední doby obsluhy (výstupu) požadavků (zákazníků).

$$\text{stř. doba vstupu} = \frac{60}{20} = 3 \text{ min}$$

$$\text{stř. doba výstupu} = 2 \text{ min}$$

$$\text{směr. odchylka výstupu} = 15 \text{ s} = 0,25 \text{ min}$$

Nyní můžeme začít zadávat hodnoty do softwaru, zvolíme exponenciální rozdělení vstupů a normální rozdělení výstupů. Jako střední dobu vstupu zadáme hodnotu 3, jako střední dobu výstupu hodnotu 2 a jako směrodatnou odchylku výstupu hodnotu 0,25. Vše ostatní ponecháme stejné jako v předchozím příkladě.

V tabulce 6 můžeme porovnat charakteristiky zkoumaného systému (stánku rychlého občerstvení) pro dobu obsluhy s exponenciálním a s normálním rozdělením pro interval 1-1000 a tisíc realizací simulace. V tabulce 7 pak střední hodnoty pravděpodobností stavů systému pro exponenciální a normální rozdělení doby obsluhy.

**Tabulka 6: Charakteristiky statického jednokan. SHO s normálním rozdělením doby obsluhy**

interval Od (min)	interval Do (min)	počet realizací	stř. počet požadavků v systému (I)		stř. doba strávená v systému (min)		stř. počet požadavků ve frontě (I)		stř. doba strávená ve frontě (min)		časová náročnost simulace (s)
			NORM	EXP	NORM	EXP	NORM	EXP	NORM	EXP	
1	1000	1000	<b>1,3369</b>	1,9704	<b>4,0106</b>	5,9113	<b>0,6714</b>	1,3065	<b>2,0142</b>	3,9196	5

**Tabulka 7: Pravděpod. stavů statického jednokan. SHO s normálním rozdělením doby obsluhy**

interval Od (min)	interval Do (min)	počet realizací	Pravděpodobnosti stavů											
			číslo stavu											
			0	1	2	3	4	5	6	7	8	9	10	
1	1000	1000												
exp. roz. doby obsluhy			0,3361	0,2240	0,1479	0,0982	0,0654	0,0430	0,0288	0,0192	0,0127	0,0084	0,0056	
norm. roz. doby obsluhy			<b>0,3345</b>	<b>0,3139</b>	<b>0,1821</b>	<b>0,0891</b>	<b>0,0425</b>	<b>0,0203</b>	<b>0,0096</b>	<b>0,0042</b>	<b>0,0020</b>	<b>0,0010</b>	<b>0,0005</b>	

Z výsledků uvedených v tabulce 6 a 7 je vidět, že výsledky jsou ovlivněny volbou statistického rozdělení.



### 5.1.2 Vícekanálový SHO

V tomto příkladu budeme simulovat pokladny v supermarketu. V supermarketu je vždy otevřeno 5 pokladen. Jedna pokladna průměrně obslouží 20 zákazníků za hodinu. V průměru přijde k pokladnám každou hodinu 80 zákazníků.

Předpokládáme, že příchody zákazníků i jejich obsluha jsou veličiny s exponenciálním rozdělením. Dále předpokládáme neomezeně dlouhou frontu typu FIFO. Pomocí Kendallovy klasifikace bychom systém popsali následovně:

$$M/M/5/\infty/FIFO$$

Nejprve si ze zadaných hodnot vyjádříme střední doby vstupu a střední doby obsluhy (výstupu) požadavků (zákazníků).

$$\text{stř. doba vstupu} = \frac{60}{80} = 0,75 \text{ min}$$

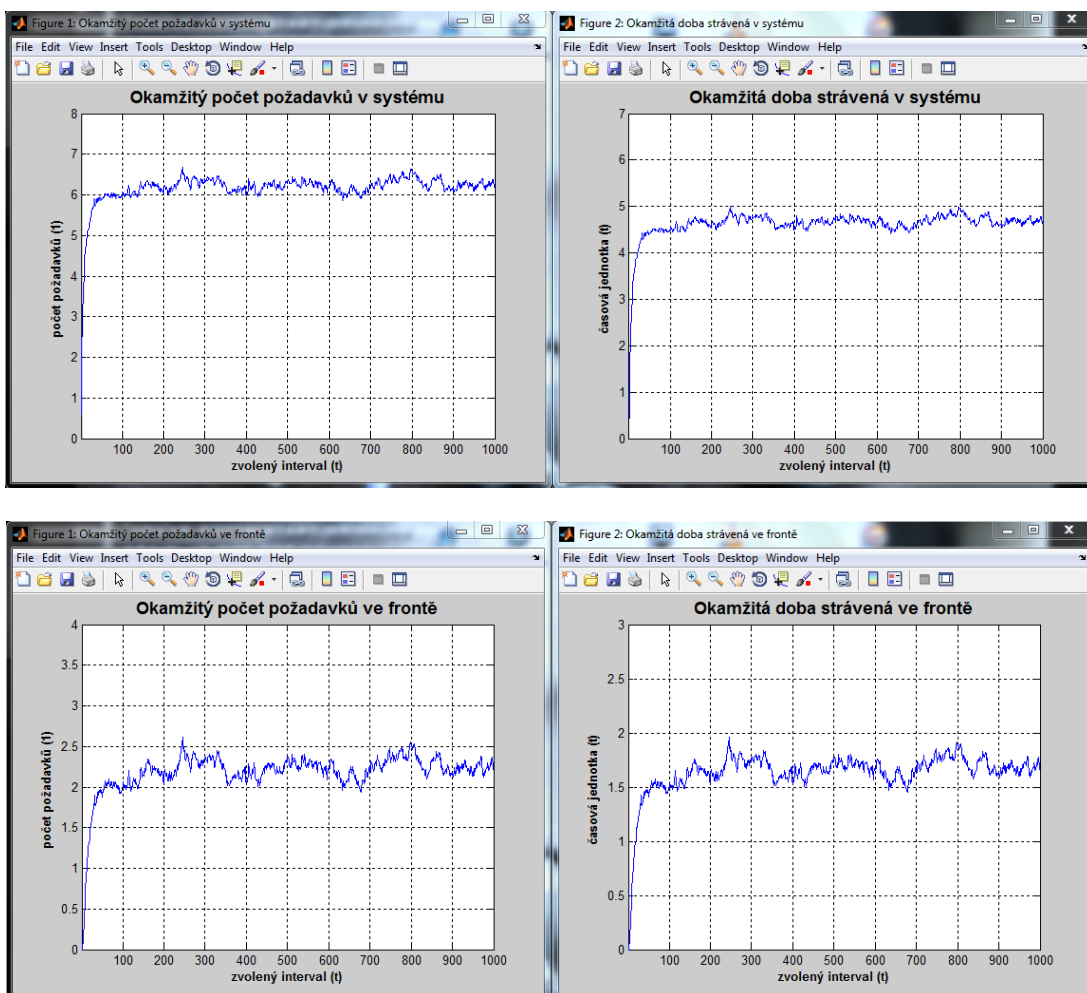
$$\text{stř. doba výstupu} = \frac{60}{20} = 3 \text{ min}$$

Nyní můžeme začít zadávat hodnoty do softwaru, zvolíme exponenciální rozdělení vstupů a výstupů. Jako střední dobu vstupu zadáme hodnotu 0.75, jako střední dobu výstupu hodnotu 3. Počet obslužných kanálů zadáme 5. Počáteční podmínky předpokládáme nulové, proto jako hodnoty počtu obsazených kanálů a počtu požadavků ve frontě zadáme 0. Maximální velikost fronty předpokládáme neomezenou, proto zvolíme vysoké číslo, např. 1000.

Budeme provádět několik simulací a ukážeme si, jak přesnosti výsledků ovlivňuje délka intervalu realizace simulace a počet realizací simulace (tabulka 8). V tabulce 8 můžeme porovnat charakteristiky zkoumaného systému (pokladen v supermarketu) získané simulací (SIM) s hodnotami získanými analyticky (ANL). Dále je v tabulce zobrazena časová náročnost simulace pro zvolené parametry simulace.


**Tabulka 8: Charakteristiky statického vícekanálového SHO**

interval Od (min)	interval Do (min)	počet realizací	stř. počet požadavků v systému (1)		stř. doba strávená v systému (min)		stř. počet požadavků ve frontě (1)		stř. doba strávená ve frontě (min)		časová náročnost simulace (s)
			SIM	ANL	SIM	ANL	SIM	ANL	SIM	ANL	
1	100	10	5,4620	<b>6,2164</b>	4,0965	<b>4,6623</b>	1,6470	<b>2,2164</b>	1,2353	<b>1,6623</b>	0,1
1	100	100	5,9696	<b>6,2164</b>	4,4772	<b>4,6623</b>	2,0951	<b>2,2164</b>	1,5713	<b>1,6623</b>	0,5
1	100	1000	5,4374	<b>6,2164</b>	4,0780	<b>4,6623</b>	1,6273	<b>2,2164</b>	1,2205	<b>1,6623</b>	2,3
1	1000	10	6,2197	<b>6,2164</b>	4,6648	<b>4,6623</b>	2,2242	<b>2,2164</b>	1,6681	<b>1,6623</b>	0,7
1	1000	100	6,0337	<b>6,2164</b>	4,5253	<b>4,6623</b>	2,0853	<b>2,2164</b>	1,5640	<b>1,6623</b>	5
1	1000	1000	6,1638	<b>6,2164</b>	4,6228	<b>4,6623</b>	2,1810	<b>2,2164</b>	1,6357	<b>1,6623</b>	44
1	1000	10000	6,1441	<b>6,2164</b>	4,6080	<b>4,6623</b>	2,1630	<b>2,2164</b>	1,6223	<b>1,6623</b>	456
1	10000	10	6,3131	<b>6,2164</b>	4,7348	<b>4,6623</b>	2,3051	<b>2,2164</b>	1,7288	<b>1,6623</b>	5
1	10000	100	6,1648	<b>6,2164</b>	4,6236	<b>4,6623</b>	2,1691	<b>2,2164</b>	1,6268	<b>1,6623</b>	45
1	10000	1000	6,1987	<b>6,2164</b>	4,6490	<b>4,6623</b>	2,2029	<b>2,2164</b>	1,6522	<b>1,6623</b>	503


**Graf 5: Charakteristiky statického vícekanálového SHO**

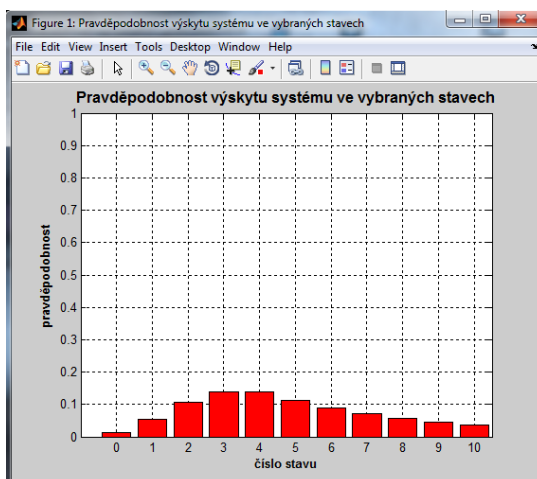


Z výsledných grafů charakteristik zkoumaného systému (Graf 5) vyplývá, že se systém ustálí po přibližně 50 minutách. Proto nejsou charakteristiky pro interval 1-100 minut tolik přesné a počet realizací nemá na výsledek příliš velký vliv. S prodloužením intervalu (pro systém, který se ustálí) a větším počtem realizací stoupá i přesnost výsledných ustálených charakteristik.

V tabulce 9 můžeme porovnat střední hodnoty pravděpodobností jednotlivých stavů získaných pomocí simulace s hodnotami získanými analyticky. V tabulce je zobrazeno prvních 11 stavů systému. Střední hodnoty pravděpodobností stavů jsou zobrazeny v grafu 6.

**Tabulka 9: Pravděpodobnosti stavů statického vícekanálového SHO**

interval Od (min)	interval Do (min)	počet realizací	Pravděpodobnosti stavů											
			číslo stavu											
			0	1	2	3	4	5	6	7	8	9	10	
analytické hodnoty			0,0130	0,0519	0,1039	0,1385	0,1385	0,1108	0,0887	0,0709	0,0567	0,0454	0,0363	
1	100	10	0,0220	0,0650	0,1170	0,1520	0,1600	0,1150	0,0900	0,0540	0,0390	0,0410	0,0360	
1	100	100	0,0223	0,0650	0,1137	0,1377	0,1375	0,1099	0,0846	0,0676	0,0515	0,0417	0,0300	
1	100	1000	0,0237	0,0675	0,1197	0,1482	0,1459	0,1123	0,0863	0,0697	0,0525	0,0415	0,0327	
1	1000	10	0,0125	0,0520	0,1085	0,1348	0,1389	0,1089	0,0886	0,0717	0,0584	0,0482	0,0355	
1	1000	100	0,0142	0,0559	0,1094	0,1436	0,1419	0,1102	0,0867	0,0691	0,0556	0,0440	0,0346	
1	1000	1000	0,0138	0,0534	0,1055	0,1392	0,1394	0,1113	0,0885	0,0707	0,0562	0,0451	0,0358	
1	1000	10000	0,0141	0,0535	0,1054	0,1397	0,1393	0,1110	0,0884	0,0707	0,0563	0,0449	0,0358	
1	10000	10	0,0129	0,0511	0,1034	0,1374	0,1376	0,1089	0,0885	0,0717	0,0568	0,0457	0,0363	
1	10000	100	0,0130	0,0522	0,1046	0,1390	0,1387	0,1108	0,0895	0,0712	0,0571	0,0456	0,0365	
1	10000	1000	0,0132	0,0521	0,1043	0,1391	0,1387	0,1107	0,0886	0,0708	0,0567	0,0453	0,0362	



**Graf 6: Pravděpodobnosti stavů statického vícekanálového SHO**





## 5.2 Dynamické SHO

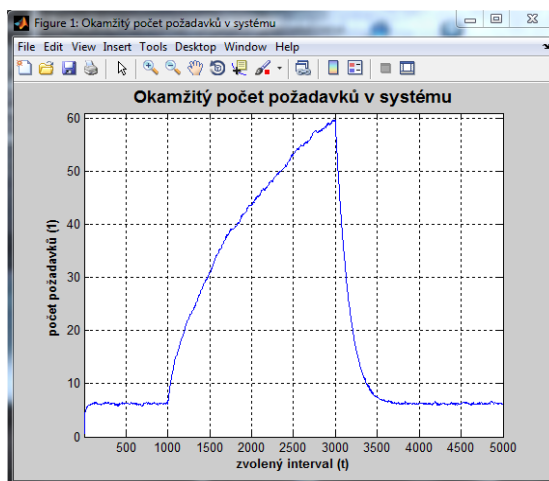
U následujících příkladů budeme demonstrovat, jak změna jednotlivých parametrů ovlivňuje chování systémů hromadné obsluhy. Konkrétně si to ukážeme na grafech středních počtů požadavků v systému. S počtem požadavků v systému/frontě se samozřejmě mění i doba strávená v systému/frontě.

U všech následujících kapitol použijeme příklad z kapitoly 5.1.2 a budeme ho modifikovat změnou parametrů. Vždy budeme provádět 1000 realizací simulace.

### 5.2.1 Změna počtu obslužných kanálů

V tomto příkladě si ukážeme, jaký má změna počtu obslužných kanálů vliv na střední počet požadavků v systému.

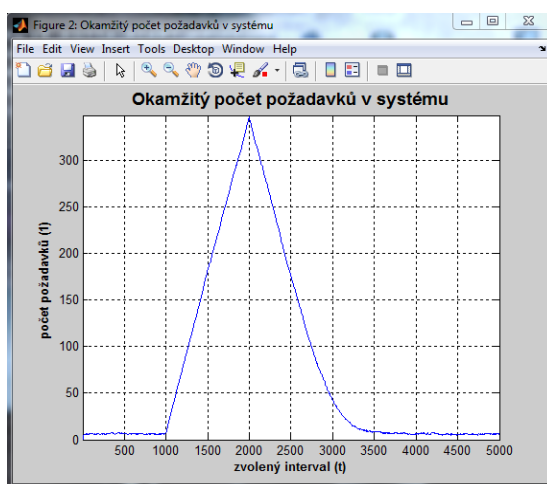
Systému z kapitoly 5.1.2 odebereme v čase 1000 minut jeden obslužný kanál (pokladnu) a v čase 3000 minut jej zase vrátíme do systému. Na grafu 7 je vidět, že v čase 1000, kdy jsme odstranili jeden obslužný kanál, se začal rapidně zvětšovat počet požadavků (zákazníků) v systému. Zbývající obslužné kanály nestačily obsluhovat příchozí požadavky. Po zapojení obslužného kanálu zpátky do systému se střední počet požadavků v systému opět vrátil na původní hodnotu.



Graf 7: Vliv změny počtu kanálů na SHO

### 5.2.2 Změna vstupního toku

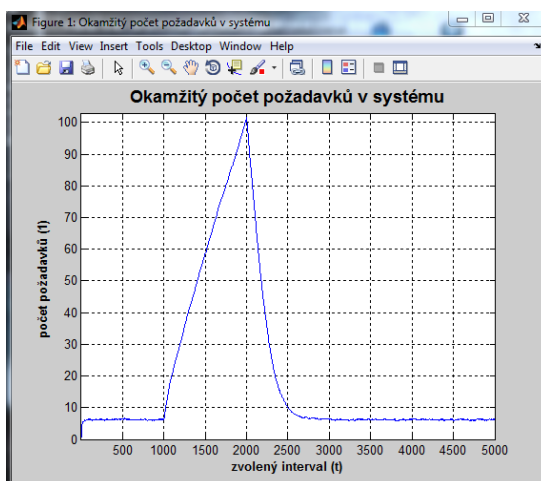
U tohoto příkladu v čase 1000 minut snížíme střední dobu vstupu požadavků (příchodů zákazníků) na 0.5 minut, tj. požadavky budou přicházet častěji. V čase 2000 minut vrátíme střední dobu vstupu požadavků na původní hodnotu 0.75. Na grafu 8 je vidět, že systém v čase 1000 přestal stačit obsluhovat příchozí požadavky a střední počet požadavků v systému začal prudce stoupat. Po vrácení parametru střední doby vstupu na předchozí hodnotu se počet požadavků v systému opět vrátil na původní hodnoty.



Graf 8: Vliv změny vstupního toku na SHO

### 5.2.3 Změna výstupního toku

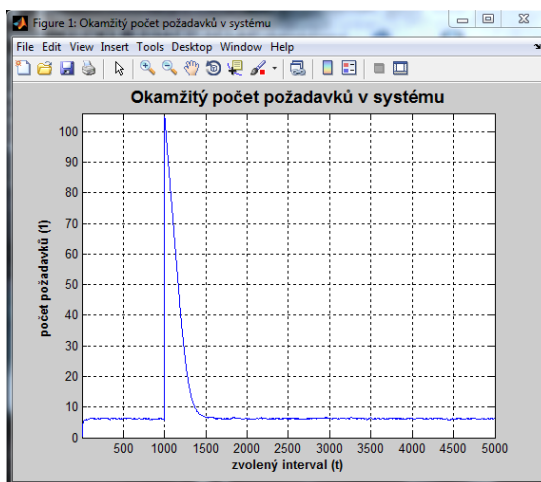
Zde v čase 1000 minut zvýšíme střední dobu výstupu požadavků (obsluhy zákazníků) na hodnotu 4 (jedna pokladna průměrně obslouží pouze 15 zákazníků). V čase 2000 minut vrátíme střední dobu výstupu požadavků na původní hodnotu 3. Graf 9 ukazuje, že obslužné kanály (pokladny) v čase 1000 přestaly stačit obsluhovat příchozí požadavky a střední počet požadavků v systému začal prudce stoupat. Po vrácení parametru střední doby výstupu na předchozí hodnotu se počet požadavků v systému opět vrátil na původní hodnoty.



**Graf 9: Vliv změny výstupního toku na SHO**

### 5.2.4 Nenulové počáteční podmínky

V tomto příkladě budeme uvažovat, že do systému přijde v čase 1000 minut najednou 100 požadavků (zákazníků). V čase 1000 minut tedy zvýšíme parametr počet požadavků ve frontě na hodnotu 100. Na grafu 10 vidíme, že se v čase 1000 minut skokově zvětšil střední počet požadavků v systému o 100. Požadavky se začaly okamžitě obsluhovat a v přibližném simulačním čase 1500 minut bylo již všech 100 požadavků obsluženo a systém se vrátil na původní hodnoty.



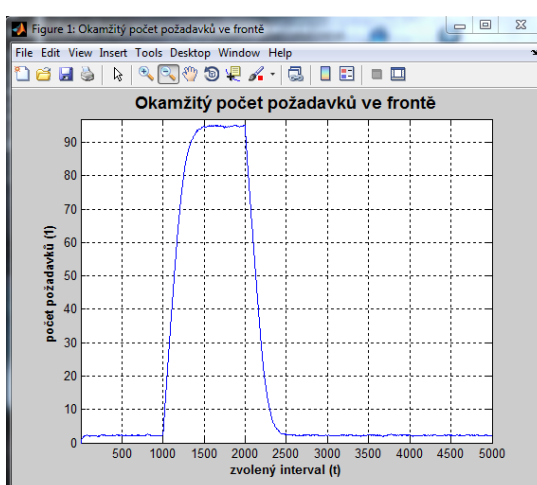
**Graf 10: Vliv změny počátečních podmínek na SHO**



### 5.2.5 Omezená velikost fronty

V posledním příkladě si ukážeme, jaký má omezená velikost fronty vliv na střední počet požadavků ve frontě.

Použijeme příklad z kapitoly 5.2.2, ale omezíme velikost fronty na 100 požadavků (zákazníků). V čase 1000 tedy zadáme jako parametr maximální velikost fronty hodnotu 100. Z grafu 11 je patrné, že když se střední počet požadavků ve frontě dostal na hodnotu 100, systém začal další nově příchozí požadavky odmítat. Z toho důvodu střední počet požadavků ve frontě nepřekročil hodnotu 100.



**Graf 11: Vliv omezení velikosti fronty na SHO**



## 6 Závěr

Vytvořený software SHO-MMC Simulator splňuje všechny požadavky, které na něj byly kladeny v kapitole 3.

Software umožňuje simulovat systémy hromadné obsluhy s exponenciálním, normálním i diskretním rozdělením dob do příchodů a odchodů požadavků. Dokáže simulovat jak statické (ustálené), tak dynamicky se měnící systémy, tj. systémy, u kterých se v čase mění vstupní/výstupní parametry (počet obslužných kanálů, střední doby příchodů a odchodů požadavků atd.). Dále umí simulovat systémy s jedním i více obslužnými kanály (s paralelním uspořádáním), nastavovat maximální velikost fronty a počáteční podmínky. Umožňuje volit dobu a počet provedení simulací.

Jako výsledky simulace dává základní charakteristiky systémů hromadné obsluhy, jako je střední počet jednotek v systému/frontě, střední doba strávená v systému/frontě, pravděpodobnost obsloužení požadavku bez čekání a pravděpodobnosti jednotlivých stavů. Dále zobrazuje grafy okamžitých a průměrných hodnot těchto charakteristik v průběhu doby simulace.

SHO-MMC Simulator je naprogramován v Matlabu a disponuje grafickým uživatelským rozhraním pro zadávání vstupních hodnot a vyhodnocování výsledků simulací.

Kapitoly 4 a 5 budou sloužit studentům jako manuál k programu SHO-MMC Simulator, protože předpokládáme jeho využití ve výuce předmětu Metody užívané v logistice (MUL).



## Literatura

- [1] Rálek P., Novák J., Chudoba J.: *Metody užívané v logistice*, Liberec: skripta TUL, 2009.
- [2] Jablonský J.: *Operační výzkum*, Praha: Vysoká škola ekonomická, 2001, 3. Vyd., 305s. ISBN 80-245-0162-7.
- [3] Žižka M.: *Vybrané statě z operačního výzkumu*, Liberec: skripta TUL, 2003, 1. Vyd., 155s. ISBN 80-7083-691-1.
- [4] Virius M.: *Aplikace matematické statistiky: metoda Monte Carlo*, Praha: skripta ČVUT 1985, 1992, 1998. 168s. ISBN 80-01-01779-6.
- [5] Fabian F., Kluiber Z.: *Metoda Monte Carlo a možnosti jejího uplatnění*, Praha: PROSPEKTRUM s.r.o., 1998. 148s. ISBN 80-7175-058-1.
- [6] Klvaňa J.: *Principy a aplikace metody Monte Carlo*, Praha: habilitační přednášky ČVUT v Praze, 2006. 34s. ISBN 80-01-03587-5.
- [7] Peško Š.: *Operační analýza 2*, skripta VŠŽ, Žilina 2001
- [8] Unčovský L.: *Stochastické modely operační analýzy*, Bratislava: Alfa Bratislava, 1. Vyd., 1980.
- [9] Šedivá a kol.: *Materiály a informace ke cvičení STAV (Výpočtová statistika)* [online], ZČU, 2008 [cit. 2011-04-17]. Dostupné z WWW: <<http://home.zcu.cz/~sediva/stav.htm>>.
- [10] Haahr M.: *Introduction to Randomness and Random Numbers*, [online], ©1998-2011, [cit. 2011-04-25]. Dostupné z WWW: <<http://www.random.org/>>.
- [11] *Random number generation* [online], poslední aktualizace 20. dubna 2011 23:20 [cit. 2011-04-25], Wikipedia. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Random\\_number\\_generation](http://en.wikipedia.org/wiki/Random_number_generation)>
- [12] Glombíková V.: *Generování náhodných čísel : Pseudonáhodná čísla* [online], 2008, [cit. 2011-05-02]. Dostupné z WWW: < [http://www.kod.tul.cz/info\\_predmety/Psi/prednasky\\_2007/prednaska\\_08/Generovan\\_08.pdf](http://www.kod.tul.cz/info_predmety/Psi/prednasky_2007/prednaska_08/Generovan_08.pdf) >
- [13] Kahánek P.: *Generátor náhodných čísel v Matlabu* [online], 2005, [cit. 2011-05-02]. Dostupné z WWW: <[http://dsp.vscht.cz/konference\\_matlab/matlab05/prispevky/kahane/kahane.pdf](http://dsp.vscht.cz/konference_matlab/matlab05/prispevky/kahane/kahane.pdf)>



- [14] Schmidt V.: *Markov Chains and Monte-Carlo Simulation* [online], University Ulm Department of Stochastic, 2006 [cit. 2011-04-25]. Dostupné z WWW: <[http://www.mathematik.uni-ulm.de/stochastik/lehre/ss06/markov/skript\\_engl](http://www.mathematik.uni-ulm.de/stochastik/lehre/ss06/markov/skript_engl)>
- [15] Šlégr J.: *Metoda Monte Carlo, Generátory náhodných čísel* [online], Univerzita Hradec Králové, PF, Katedra fyziky a informatiky, 2007 [cit. 2011-04-25]. Dostupné z WWW: <<http://www.black-hole.cz/soubory/mc.pdf>>
- [16] Zukerman M.: *Introduction to Queueing Theory and Stochastic Teletraffic Models* [online], ©2000-2011, 165s. [cit. 2011-04-15]. Dostupné z WWW: <<http://www.ee.cityu.edu.hk/~zukerman/classnotes.pdf>>
- [17] Iversen V.: *Teletraffic Engineering Handbook* [online], 2001, 310s. [cit. 2011-04-15]. Dostupné z WWW: <[http://www.itu.int/ITU-D/study\\_groups/SGP\\_1998-2002/SG2/StudyQuestions/Question\\_16/RapporteursGroupDocs/teletraffic.pdf](http://www.itu.int/ITU-D/study_groups/SGP_1998-2002/SG2/StudyQuestions/Question_16/RapporteursGroupDocs/teletraffic.pdf)>
- [18] Kubát J.: *Simulace procesů hromadné obsluhy na číslicových počítačích*, Praha, Institut manipulačních a dopravních systémů, 1970. 228s.
- [19] Zaplatílek K., Doňar B.: *Matlab – tvorba uživatelských aplikací*, Praha: BEN – technická literatura, 2005. 216 s. ISBN 80-7300-133-0.
- [20] Bartko R., Miller M.: *Matlab I. – algoritmizácia a riešenie úloh*, Trenčín: DIGITAL GRAPHICS, 2002. 288s. ISBN 80-968337-3-1.
- [21] *Box-Muller transform* [online], poslední aktualizace 3. dubna 2011 00:08 [cit. 2011-04-25], Wikipedia. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Box%E2%80%93Muller\\_transform](http://en.wikipedia.org/wiki/Box%E2%80%93Muller_transform)>
- [22] Blinkdagger.: *Matlab GUI Tutorials* [online], ©2011 [cit. 2011-02-10]. Dostupné z WWW: <<http://blinkdagger.com/matlab>>
- [23] Mathworks.: *Matlab* [online], ©1994-2011 The Mathworks, Inc [cit. 2011-05-02]. Dostupné z WWW: <<http://www.mathworks.com/products/matlab/>>
- [24] *Exponenciální rozdělení* [online], poslední aktualizace 17. února 2011 10:59 [cit. 2011-05-10], Wikipedie. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Exponenci%C3%A1ln%C3%AD\\_rozd%C4%9Blen%C3%AD](http://cs.wikipedia.org/wiki/Exponenci%C3%A1ln%C3%AD_rozd%C4%9Blen%C3%AD)>
- [25] *Internetová prezentace společnosti Simul8 Corporation* [online] [cit. 2011-05-05]. Dostupné z WWW: <<http://www.simul8.cz>>



- [26] *Internetová prezentace softwaru Goldsim* [online] [cit. 2011-05-05]. Dostupné z WWW: <<http://goldsim.com/Home>>
- [27] *Internetová prezentace softwaru Arena* [online], [cit. 2011-05-05]. Dostupné z WWW: <<http://www.arenasimulation.com>>
- [28] *Internetová prezentace společnosti Humusoft s.r.o.* [online], [cit. 2011-05-05]. Dostupné z WWW: <<http://www.humusoft.cz/witness/witnescz.htm>>
- [29] *Internetová prezentace společnosti Lanner Group Limited* [online] [cit. 2011-05-05]. Dostupné z WWW: <<http://www.lanner.com>>
- [30] *Internetová prezentace společnosti ProModel Corporation* [online] [cit. 2011-05-05]. Dostupné z WWW: <<http://www.promodel.com>>





## Seznam obrázků

Obrázek 1: Obecné schéma systému hromadné obsluhy .....	12
Obrázek 2: Obecné schéma SHO s uspořádáním obslužných kanálů .....	17
Obrázek 3: Tabulka náhodných čísel .....	31
Obrázek 4: Výstupy z kongruentních gen. při nevhodně zvolených konstantách .....	33
Obrázek 5: Příklad opravdu náhodně rozložených bodů v prostoru .....	34
Obrázek 6: Ukázka programového prostředí Matlab .....	38
Obrázek 7: Software SHO-MMC Simulator .....	44

## Seznam grafů

Graf 1: Distribuční funkce rozdělení náhodné veličiny .....	36
Graf 2: Nespojitosť v grafu doby strávené v systému u dynamického SHO .....	52
Graf 3: Charakteristiky statického jednobanálního SHO .....	59
Graf 4: Pravděpodobnosti stavů statického jednobanálního SHO .....	60
Graf 5: Charakteristiky statického vícebánálního SHO .....	63
Graf 6: Pravděpodobnosti stavů statického vícebánálního SHO .....	64
Graf 7: Vliv změny počtu kanálů na SHO .....	65
Graf 8: Vliv změny vstupního toku na SHO .....	66
Graf 9: Vliv změny výstupního toku na SHO .....	67
Graf 10: Vliv změny počátečních podmínek na SHO .....	67
Graf 11: Vliv omezení velikosti fronty na SHO .....	68

## Seznam tabulek

Tabulka 1: Ukázka matice Generátor .....	41
Tabulka 2: Ukázka matice Realizace .....	42
Tabulka 3: Ukázka matice Stav .....	42
Tabulka 4: Charakteristiky statického jednobanálního SHO .....	58
Tabulka 5: Pravděpodobnosti stavů statického jednobanálního SHO .....	60
Tabulka 6: Charakteristiky statického jednoban. SHO s normálním rozdělením doby obsluhy .....	61
Tabulka 7: Pravděpod. stavů statického jednoban. SHO s normálním rozdělením doby obsluhy .....	61
Tabulka 8: Charakteristiky statického vícebánálního SHO .....	63
Tabulka 9: Pravděpodobnosti stavů statického vícebánálního SHO .....	64



## Seznam symbolů a zkratek

$a$	násobitel
$c$	inkrement
$F$	distribuční funkce
$F^{-1}$	inverzní funkce k distribuční funkci
$GUI$	Graphical User Interface
$GUIDE$	Graphical User Interface Development Environment
$\lambda$	střední intenzita vstupního toku
$M$	modul
$MCR$	Matlab Compiler Runtime
$\mu$	střední intenzita výstupního toku
$mod$	modulo, zbytek po celočíselném dělení
$N(\mu, \sigma^2)$	normální rozdělení s parametry $\mu, \sigma^2$
$\bar{n}_f$	střední počet požadavků ve frontě
$\bar{n}_s$	střední počet požadavků v systému
$p_n$	pravděpodobnost stavu $n$
$PRNG$	Pseudo-Random Number Generator
$prs$	pravděpodobnost
$rand$	náhodné číslo z intervalu (0,1)
$\rho$	střední intenzita provozu
$\bar{t}_f$	střední doba strávená ve frontě
$\bar{t}_s$	střední doba strávená v systému
$TRNG$	True Random Number Generator
$u_1, u_2$	hodnoty náhodné veličiny s normovaným normálním rozdělením
$x$	počet obslužných kanálů



## Seznam příloh

Příloha 1: Obsah CD .....	76
---------------------------	----



## Obsah CD

- DP\_Tomas\_Masopust.pdf – text diplomové práce ve formátu pdf
- SHO\_MMC\_Simulator32.exe – simulační software SHO-MMC Simulator pro Win32
- SHO\_MMC\_Simulator64.exe – simulační software SHO-MMC Simulator pro Win64
- MCRIInstaller.exe – soubory sdílených knihoven Matlabu pro Win32 a Win64